

Ellipse Detection via Global Arc Compatibilities and Adaptive Co-Clustering for Real-World Measurement Systems

Zihan Wu

zihan.wu@my.cityu.edu.hk

City University of Hong Kong <https://orcid.org/0000-0002-6551-6177>

Zhaoke Huang

City University of Hong Kong

Hong Yan

City University of Hong Kong <https://orcid.org/0000-0001-9661-3095>

Research Article

Keywords: Co-clustering, computer vision, ellipse detection, image processing, Singular Value Decomposition (SVD)

Posted Date: March 14th, 2025

DOI: <https://doi.org/10.21203/rs.3.rs-3926094/v2>

License:   This work is licensed under a Creative Commons Attribution 4.0 International License.
[Read Full License](#)

Additional Declarations: The authors declare no competing interests.

Ellipse Detection via Global Arc Compatibilities and Adaptive Co-Clustering for Real-World Measurement Systems

Zihan Wu, Zhaoke Huang, Hong Yan, *Fellow, IEEE*

Abstract—Ellipse detection plays a crucial role in computer vision and measurement systems, yet existing methods often struggle with handling spatially distant arc segments, high computational complexity, and sensitivity to noise and complex geometries. In this paper, we propose a novel ellipse detection method that addresses these limitations through global arc compatibility analysis and adaptive co-clustering. We develop a probabilistic framework to represent ellipse distributions associated with detected arc segments, using Jensen-Shannon (JS) divergence to quantify arc compatibilities across the entire image. Moreover, we design an adaptive co-clustering algorithm based on Singular Value Decomposition (SVD) that efficiently groups compatible arcs while automatically estimating the number of ellipses present. This approach significantly improves detection accuracy in complex scenarios, as demonstrated by our experiments on synthetic and real-world datasets, including applications in industrial quality control and optical calibration systems. Compared to state-of-the-art methods, our algorithm achieves a 10% improvement in accuracy and 15% in recall on synthetic images, excelling particularly in detecting small and occluded ellipses. Our method provides a robust, efficient solution for ellipse detection, enhancing measurement precision across diverse imaging conditions.

Index Terms—Co-clustering, computer vision, ellipse detection, image processing, Singular Value Decomposition (SVD)

I. INTRODUCTION

Ellipse detection is a fundamental task in computer vision and pattern recognition with extensive applications across various fields, including measurement systems for industrial automation, medical imaging, and other related areas [1], [2], [3]. In industrial quality control, ellipse detection helps monitor circular components, ensuring product integrity and enabling real-time feedback on manufacturing processes [4]. Autonomous systems, such as self-driving vehicles, rely on robust ellipse detection for identifying road signs and traffic lights, contributing to safety and navigation [5], [6]. In medical imaging, robust ellipse detection techniques are essential for identifying anatomical structures, including blood cells, tumors, and eye features [7], [8], [9].

Existing ellipse detection methods face three major challenges. First, many traditional approaches, particularly edge-following methods [10], [11], [12], [13], [14], [15], rely heavily on local features and spatial proximity. While efficient, these methods struggle to associate arc segments belonging to the same ellipse but spatially distant in the image, and they

perform poorly when dealing with partially occluded or large ellipses. Second, methods based on the Hough Transform (HT) [16], [17], a feature extraction technique that identifies parametric shapes by transforming image points into a parameter space where shapes are detected as peaks in an accumulator array, have high computational complexity, especially when processing complex scenes containing multiple ellipses. This leads to a significant increase in computation time and memory usage. Moreover, these methods often depend on preset thresholds or fixed parameters, making it difficult to adapt to images of varying complexity. For instance, Randomized Hough Transform (RHT) proposed by McLaughlin [18] requires careful parameter tuning for different scenarios. Third, while least squares methods [19] are computationally efficient, they are highly sensitive to noise and cluttered backgrounds and typically employ single estimates for ellipse fitting. This approach leads to false detections or missed detections, reducing the reliability of detection, particularly in complex real-world scenarios with multiple overlapping ellipses or significant background clutter. These limitations reduce the reliability and scalability of ellipse detection in practical measurement applications.

To address these limitations, we propose a novel ellipse detection method based on global arc compatibility and adaptive co-clustering. Our method introduces three key innovations. First, we develop a global arc compatibility analysis framework that uses a probabilistic representation for each detected arc segment, showing the range of ellipses it might belong to. This probabilistic representation captures the uncertainty in arcs and allows for a more flexible association between arcs and ellipses. Jensen-Shannon (JS) divergence [20], an information-theoretic measure of similarity between probability distributions, is then employed to quantify the compatibility between the probabilistic distributions of different arcs, regardless of their spatial distance. This global analysis allows for the effective association of arc segments, improving robustness in detecting large, partially occluded, or fragmented ellipses.

Second, we propose an adaptive co-clustering algorithm that efficiently groups arc segments into potential ellipses. Our method applies Singular Value Decomposition (SVD) [21] to the arc compatibility matrix, projecting the high-dimensional compatibility matrix into a lower-dimensional space, preserving the most significant arc relationships while reducing noise. This dimensionality reduction enhances computational efficiency and makes the subsequent clustering more robust to noise and occlusions.

Third, we design a novel method for adaptive ellipse number estimation. By analyzing the cumulative energy distribution of singular values from the SVD of the compatibility matrix, we

Zihan Wu, Zhaoke Huang, and Hong Yan are with the Department of Electrical Engineering, City University of Hong Kong, Hong Kong, China (e-mail: zihan.wu@my.cityu.edu.hk). Corresponding author: Zihan Wu.

This paper has been accepted for publication in *IEEE Transactions on Instrumentation and Measurement*. The final version will be available at IEEE Xplore upon publication. Please refer to the published version once available.

determine the number of ellipses present in the image without manual tuning. This makes our method highly adaptable to varying scene complexities, ensuring effective performance in diverse real-world environments.

Our main contributions can be summarized as follows:

- 1) We propose a novel ellipse detection method based on global arc compatibility and adaptive co-clustering. This approach introduces a global perspective on arc relationships, enabling the association of arc segments belonging to the same ellipse even when they are spatially distant in the image.
- 2) We develop a probabilistic framework for representing ellipse distributions associated with detected arc segments. Using JS divergence [20] to measure distribution compatibility, our method robustly quantifies global arc relationships, enhancing resilience to noise and complex geometries.
- 3) We design an adaptive co-clustering algorithm based on SVD for efficient arc grouping and automatic ellipse number estimation. This approach leverages dimensionality reduction to preserve significant compatibility information while eliminating manual parameter tuning, thereby improving efficiency and accuracy.

The rest of the paper is organized as follows: Section II reviews related works in ellipse detection, highlighting the strengths and limitations of existing approaches, and discusses co-clustering techniques. Section III details our proposed method, describing the co-clustering framework and how it addresses the challenges in ellipse detection. Section IV presents experimental results and analysis, comparing our method against state-of-the-art approaches. Finally, Section V concludes the paper and discusses future research directions.

II. RELATED WORKS

A. Ellipse Fitting Based Methods

The least-square approach, developed as the first ellipse fitting method, is still relevant due to its closed-form solutions and limited computational resource requirements. However, its simplicity renders it susceptible to noise and the constraint of fitting only one ellipse at a time [19]. Algebraic methods may produce biased estimations, with high curvature bias posing a significant issue [22]. A non-iterative least-square algorithm was proposed for ellipse-specific fitting, which utilizes the normalization of $4ac - b^2 = 1$, where a , b , and c are coefficients in the general conic equation $ax^2 + bxy + cy^2 + dx + ey + f = 0$. This normalization constraint ensures that the fitted curve is an ellipse (rather than another conic section) while minimizing the total algebraic distance [23]. This method remains the most popular choice for ellipse fitting. Recently, an ensemble-based approach has been introduced that combines multiple ellipse fitting methods to improve accuracy, especially in noisy or occluded scenarios. This method employs a consensus mechanism in a specially chosen parameter space and utilizes the L1 median for robustness against defective fits [24]. Additionally, a new efficient method that uses the k -means algorithm with Mahalanobis circles as cluster centers has been proposed for the multiple ellipse detection problem [25]. However, these

methods mostly focus on local optimizations and individual ellipse fitting, which limits their ability to address occlusions and overlapping arcs effectively.

B. Hough Transform Based Methods

The Hough Transform (HT) was introduced as a means to estimate ellipse parameters through a voting scheme [16]. Each edge pixel votes within a 5D-parameter space F , and the peaks in F are selected as ellipses. Although simple to implement, HT is computationally expensive in terms of storage and time costs, rendering it unsuitable for direct ellipse detection [17]. In addressing these limitations, Premachandra *et al.* [26] proposed a speed-up ellipse enclosing character detection algorithm, significantly reducing processing time by employing parallel image scanning and selective application of HT on large-size document images. Randomized Hough Transform [27] and Probability Hough Transform (PHT) [28] were subsequently proposed, transferring the voting process to a many-to-one style. Randomized Hough Transform randomly selects three non-collinear points and votes for the corresponding ellipse, performing a complete workflow. However, it is sensitive to occlusion and overlapping ellipses [18]. Several optimizations to the Hough Transform have been proposed. Xie *et al.* [29] reduced the five-dimensional parameter space to one dimension by estimating the semi-axis length of hypothetical ellipses, while Chia *et al.* [30] exploited ellipse foci properties to transform the five-dimensional space into a two-dimensional one, using a one-dimensional accumulator to store votes. Despite these optimizations, HT-based methods are generally inadequate for images with occlusions and noise, especially when specific thresholds are not provided [13].

C. Edge Following Methods

Edge following focuses on locally connecting detected arcs based on their geometric properties to form an ellipse. A method was developed to extract short straight lines, approximate them by elliptical arcs, and connect them to form an ellipse [10]. This method was optimized to consume less memory [31] and improved by linking line segments based on local geometric properties, such as adjacency and curvature constraints [11]. A recent approach proposes a novel ellipse detection method utilizing coherent chord computation and projective invariant cross ratio, achieving high accuracy and robustness in complex images [15]. Another method improves both accuracy and computational efficiency by using a quadrant representation and a top-down fitting strategy [32]. However, edge-following methods heavily rely on local properties, which limits their effectiveness in handling discontinuities, occlusions, or widely separated arcs.

Various other methods have been proposed to enhance ellipse detection, including using the HT to estimate parameters while decomposing the voting space into smaller subspaces [33], [34], introducing a pruning strategy to reduce the number of arcs considered [12], dividing arc candidates into quadrants for better grouping performance [35], employing line detection methods for high-quality ellipse detection [36], [13], utilizing Arc Adjacency Matrices to pair arcs by encoding

curvatures and tangents [37], and incorporating convex-hulls into the detection procedure to distinguish arc distance, length, direction, and convexity, thereby improving performance [14]. While these methods introduce various optimizations, they still largely depend on local features. Recent work has also explored multi-task learning frameworks that combine ellipse localization, contour detection, and parameter prediction to improve accuracy and robustness, though with a slight increase in computation time [38].

D. Co-Clustering Techniques

Co-clustering, also known as biclustering for 2D data, was initially developed for gene expression data analysis [39]. Additional algorithms soon emerged that were specifically tailored for biological data [40]. A significant development came in the form of a greedy-based algorithm enriched with a local search strategy, designed to navigate effectively through poor local minima [41]. The technique has also been applied in collaborative filtering [42], and its utility has been demonstrated in text mining [43] and image analysis [44]. Recent advances in the field also propose a multi-scale co-clustering approach for tensor data, which leverages tensor and matrix decomposition to identify coherent patterns along multiple dimensions [45]. Building on recent advancements, the study by Wei *et al.* [46] introduces a hierarchical high-order co-clustering algorithm that maximizes modularity. This method effectively merges multiple feature spaces for co-clustering star-structured high-order heterogeneous data.

III. PROPOSED METHOD

In this section, we describe our proposed co-clustering-based ellipse detection method in detail. The approach involves several key steps: extracting arcs from the input image, assessing arc compatibility through co-clustering, and validating the detected ellipses. We start with an overview of our approach, followed by a detailed explanation of how arcs are detected and grouped. We then discuss the co-clustering process and the details of the implementation that enhance performance. This structure ensures a clear understanding of our novel ellipse detection framework. The flowchart of our method is shown in Figure 1.

A. Method Overview

Our proposed ellipse detection method leverages a novel distribution co-clustering framework to effectively detect ellipses in both synthetic and real-world images. The method begins with elliptical arc detection, employing a robust two-step approach. First, we generate arc candidates through gradient analysis and region growing. Then, we validate these candidates using the *a contrario* framework [48], which utilizes the Number of False Alarms (NFA) to assess statistical significance. These segments are then merged to form candidate elliptical arcs, which are subsequently validated using the *a contrario* framework [36], [48].

Following the elliptical arc detection, we compute global arc compatibility using a probabilistic approach. The ellipse

each arc belongs to is represented as a probability distribution, allowing us to compute overlap compatibility scores using the JS divergence [20]. This process culminates in the construction of a global compatibility matrix, which encapsulates the compatibilities between all detected arcs.

The final stage of our method involves arc co-clustering, implemented through an SVD-based framework. We perform SVD on the normalized compatibility matrix and project the data onto a lower-dimensional space. The *k*-means clustering [47] is then applied to group arcs belonging to the same ellipse. Importantly, we estimate the number of ellipses using the cumulative energy of singular values, providing an adaptive approach to varying image complexities.

Our algorithm's central premise is the accurate clustering of candidate arcs that may be far apart but constitute the same ellipse. This approach addresses key challenges in ellipse detection: computational complexity is significantly reduced, the number of ellipses is adaptively estimated, and long-range dependencies between arcs are captured, enabling the detection of partially occluded or large ellipses. Moreover, the global perspective afforded by our method allows for robust detection in the presence of noise and clutter.

By addressing common challenges such as occlusions, overlaps, and noise, our comprehensive approach enables accurate and efficient ellipse detection across a wide range of applications. The method's ability to handle complex scenarios and its computational efficiency represent significant advancements in the field of ellipse detection.

B. Elliptical Arc Detection

Elliptical arc detection is crucial for ellipse detection. Thus, a robust arc detection method for reliably identifying elliptical arcs in input images is introduced in this paper. The approach consists of two main steps: candidate generation and the *a contrario* validation [48], [36].

The first step involves finding candidate elliptical arcs. This is done by smoothing the input image and then analyzing the gradient information to identify regions with similar gradient characteristics. The process starts with a high-gradient seed pixel and gradually adds connected pixels with similar gradient orientation and strength until a complete region is formed. This region is considered a candidate arc segment. Then, the algorithm merges rectangles with common endpoints to identify elliptical arcs. This process continues until no further merging is possible and the resulting merged arc segments are labeled as candidate elliptical arcs.

Once candidate arcs are generated, the *a contrario* framework is applied to evaluate their statistical significance. This framework utilizes the NFA to assess the likelihood that a detected arc could have occurred by chance in a random image. A lower NFA indicates higher statistical significance. Given an image x and a random image X from the *a contrario* model H_0 , the NFA for a candidate elliptical arc a_x is defined as:

$$\text{NFA}(a_x) = N_a \cdot P[\tilde{k}(a_X) \leq \tilde{k}(a_x)], \quad (1)$$

where N_a is the number of possible elliptical arc candidates in the image, $\tilde{k}(a_x)$ is a measure of structuredness for the

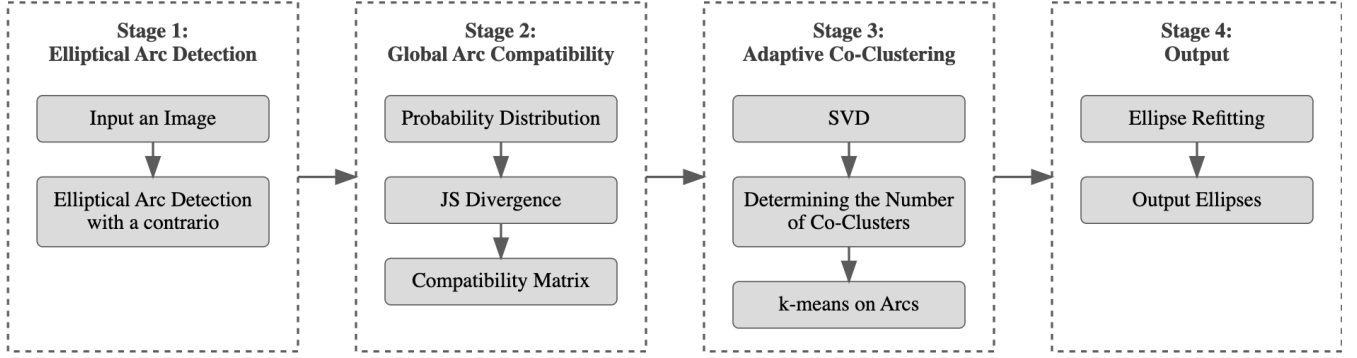


Fig. 1: Workflow of the proposed ellipse detection method. The process consists of four main stages: (1) Elliptical Arc Detection: The input image undergoes arc detection using an *a contrario* approach [36]. (2) Global Arc Compatibility Computation: Detected arcs are analyzed through probability distribution calculation, followed by JS divergence [20] computation to construct a global compatibility matrix. (3) Adaptive Co-Clustering: The compatibility matrix is subjected to SVD, followed by determining the optimal number of co-clusters, and then *k*-means clustering [47] is applied to group compatible arcs. (4) Output: The clustered arcs undergo ellipse refitting to produce the final set of detected ellipses. This workflow combines local arc information with global compatibility computation to achieve robust ellipse detection.

arc a_x in the observed image, and $P[\tilde{k}(a_x) \leq \tilde{k}(a_x)]$ is the probability of observing an arc as structured as a_x in a random image. The probability term is computed using the Irwin-Hall distribution, as the structuredness measure is based on the sum of angular differences between the arc's expected gradient direction and the observed gradient directions. A candidate arc a_x is considered valid if its NFA is below a predefined threshold ε (typically set to 1):

$$\text{NFA}(a_x) \leq \varepsilon. \quad (2)$$

This NFA formulation automatically adjusts detection thresholds to balance false positives and sensitivity to true geometric structures. By setting $\varepsilon = 1$, the method ensures that on average, at most one false detection is expected in a random image, providing a principled way to control false detections while remaining sensitive to true elliptical arcs.

To ensure robust and accurate arc detection while avoiding local minima, we incorporate three interconnected strategies into our approach. First, seed points are carefully selected during the candidate generation step by prioritizing pixels with strong gradient magnitudes, focusing on regions with significant edge information that are likely to correspond to meaningful arc structures. Among these, we choose the seed with the maximum gradient strength to initiate the region growing process, ensuring a reliable and meaningful starting point. Second, during region growing, geometric constraints such as convexity and smoothness are enforced, preventing the process from extending into noisy or irregular areas. Once these constraints are violated, the growth process will be terminated. Third, the validation step leverages the NFA criterion, selecting the candidate arc with the smallest NFA when multiple options exist to ensure the most statistically significant and meaningful detection. By integrating informed seed selection, constrained region growth, and rigorous statistical validation, our method achieves robust elliptical arc detection while minimizing the risk of false detections or entrapment in local minima.

C. Global Arc Compatibility Computation

After obtaining elliptical arcs from the first step, our method proceeds to analyze the global compatibility of these arcs, laying the groundwork for the final co-clustering stage where compatible arcs are grouped into complete ellipses. This second step of computing arc compatibilities serves as a crucial bridge, transforming local arc information into a global representation that enables effective ellipse detection even in complex scenarios.

Traditional ellipse detection methods often rely on heuristic thresholds and local geometric properties. They struggle to accurately detect ellipses in complex scenarios and recognize distant arcs as potentially belonging to the same ellipse, leading to incomplete or fragmented detections. To overcome these challenges and achieve more robust and accurate ellipse detection, we propose a global arc compatibility computation method based on probabilistic modeling.

Unlike traditional methods that fit a single ellipse to a group of arcs, we adopt a probabilistic approach where the true ellipse associated with an arc is represented by a distribution rather than a single estimate. While probabilistic approaches have been previously used in curve detection [49], [50], [51], our method represents the first application of JS divergence for measuring elliptical arc distributions. Previous methods typically relied on geometric properties [11] or voting schemes [27]. Our probabilistic framework offers several advantages: it naturally handles uncertainty in arc detection, provides a principled way to measure arc compatibility regardless of spatial distance, and enables global optimization through the compatibility matrix.

In our framework, the fitted ellipse corresponds to the maximum likelihood estimate, capturing the inherent uncertainty in arc detection and ellipse fitting. The key idea is that arcs belonging to the same ellipse will exhibit high probability densities in a specific region of the ellipse parameter space.

To formalize this, we define a probability distribution $P(E|A)$ for detecting an ellipse E given an arc A , where E is a stochastic variable representing the ellipse and A is a stochastic variable representing the arc. Using Bayes'

theorem [52], we have:

$$P(E|A) = \frac{P(A|E)P(E)}{P(A)}, \quad (3)$$

where $P(A|E)$ is the probability of observing arc A given ellipse E , $P(E)$ is the prior probability of an ellipse, and $P(A)$ is the probability of detecting the arc. This formulation follows from Bayes' theorem [53]. When implementing Equation (3), we introduce a threshold-based approach to ensure numerical stability:

$$\begin{aligned} \text{if } P(A) < \tau_{\text{threshold}} : & \text{ skip current computation} \\ \text{else : } & P(E|A) = \frac{P(A|E)P(E)}{P(A)} \end{aligned} \quad (4)$$

where $\tau_{\text{threshold}} = 1 \times 10^{-6}$. This approach is both mathematically sound and computationally efficient: when $P(A)$ falls below the threshold, it indicates that the arc detection probability is negligibly small, making it safe to skip the corresponding probability computation. By avoiding calculations with extremely small denominators, this simple strategy prevents numerical instability while preserving the method's effectiveness.

To calculate the concrete probability distribution of an ellipse given an arc, consider a grayscale image $\mathbf{I} \in [0, 1]^{H \times W}$, where H and W represent the height and width of the image in pixels, and a candidate arc a with a supporting point set S_a .

$$\mu_a(e) = p(e|S_a) = \frac{\prod_{i=1}^N \nu_e(q_i)}{\int \prod_{i=1}^N \nu_{e'}(q_i) de'}, \quad (5)$$

where $\nu_e(q)$ denotes the probability density of a point $q = (x_q, y_q) \in S_a$ supporting e , and N is the number of supporting pixels.

Using these distributions, we quantify the likelihood of two arcs belonging to the same ellipse based on the overlap of their respective ellipse distributions. For arcs a_1 and a_2 with distributions $\mu_{a_1}(e)$ and $\mu_{a_2}(e)$, we define an overlap compatibility score:

$$S(a_1, a_2) = \int_e \min(\mu_{a_1}(e), \mu_{a_2}(e)) de. \quad (6)$$

This score, a Szymkiewicz-Simpson coefficient [54], represents the overlapping area of the two distributions in the ellipse parameter space. A high value indicates a high probability that a_1 and a_2 belong to the same ellipse.

To efficiently compute this overlap compatibility score, we use the JS divergence [20], which is symmetric and non-negative, making it particularly suitable for assessing compatibilities between pairs of arcs. Lower JS divergence values indicate greater compatibilities. The JS divergence, first introduced by Lin [20], is defined as:

$$\text{JS}(P \parallel Q) = \frac{1}{2} \text{KL}(P \parallel M) + \frac{1}{2} \text{KL}(Q \parallel M), \quad (7)$$

where KL denotes the Kullback-Leibler divergence [55], and M is the average of P and Q .

Using the JS divergence [20], we construct a compatibility matrix $\mathbf{B}$, where each element $\mathbf{B}_{ij}$ is computed as:

$$\mathbf{B}_{ij} = \text{JS}(\mu_{a_i} \parallel \mu_{a_j}). \quad (8)$$

This symmetric compatibility matrix encapsulates the compatibility between each pair of arcs, providing a global view of arc compatibilities.

The primary advantage of our approach lies in its global perspective, which is particularly effective for handling partially occluded ellipses or those with spatially distant arc segments. Unlike traditional methods that rely on local spatial proximity or geometric properties, our method operates in the ellipse parameter space, allowing us to correlate arcs based on their distributions, irrespective of their spatial distances in the image.

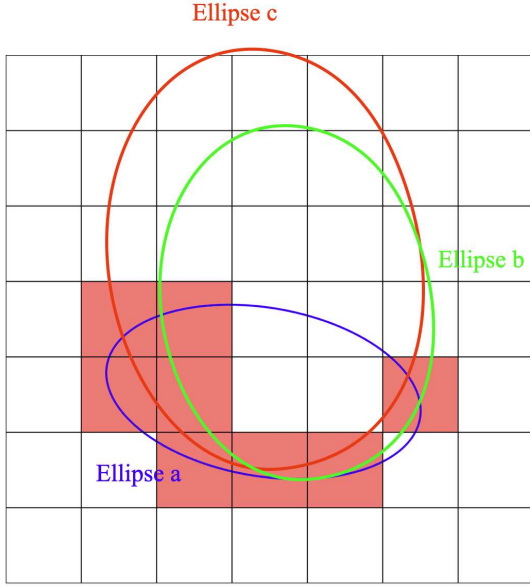
This global perspective is especially valuable when dealing with occluded ellipses, large ellipses with widely separated arc segments, and images with significant noise or clutter. By considering the overall distribution of possible ellipses for each arc, our method can correlate visible segments even when they are not contiguous in the image space.

For instance, in Figure 3, the red arc a_1 and the blue arc a_3 form one ellipse, while the yellow arc a_2 and the green arc a_4 form another. Although the blue arc a_3 is spatially close to both the red and yellow arcs, our method relies on the overlap of the distributions $\mu_{a_i}(e)$ rather than proximity. The compatibility score $S(a_1, a_3)$ would be high, indicating strong compatibilities for forming a vertically oriented ellipse, while $S(a_2, a_4)$ similarly indicates a horizontally oriented ellipse. In contrast, scores like $S(a_1, a_2)$ or $S(a_3, a_4)$ would be lower, reflecting less compatibilities.

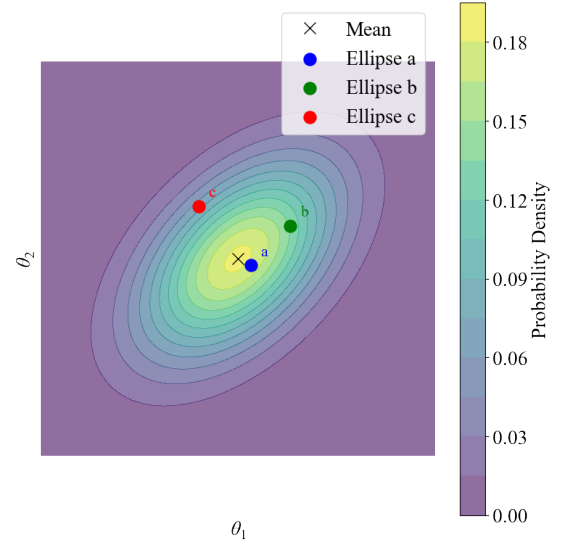
Supporting pixels are discrete pixels identified in the input image that collectively form a potential elliptical arc pattern (see Figure 2a). These pixels serve as the foundational evidence for fitting an elliptical arc. For each supporting pixel q_i , we compute the probability density $\nu_e(q_i)$ that it belongs to a candidate ellipse e . This density is determined by considering both the geometric distance from the pixel to the ellipse and how well the pixel aligns with the grid structure. The complete set of these supporting pixels S_a forms the basis for computing the overall probability distribution $\mu_a(e)$ over the space of possible ellipse parameters (Figure 2b), where each candidate ellipse is evaluated based on how well it fits all supporting pixels collectively.

This approach effectively captures long-range dependencies between arcs, enabling accurate and robust ellipse detection even in complex scenarios where local spatial cues might be misleading. The global perspective of our method allows it to handle challenging configurations and provides comprehensive and reliable ellipse detection in diverse image conditions.

The compatibility matrix $\mathbf{B}$ constructed through JS divergence computations is not necessarily always sparse, as it captures meaningful relationships between all pairs of validated arcs. The SVD-based dimensionality reduction is employed primarily to enhance computational efficiency and extract the most significant patterns from this relationship matrix, rather than addressing sparsity concerns.



(a) Supporting pixels (red) and three possible elliptical arc fits based on these supporting pixels.



(b) 2D projection of ellipse parameter probability distribution $\mu_a(e)$.

Fig. 2: Illustration of elliptical fitting and its probability distribution: (a) A set of supporting pixels (shown in red) with their three possible elliptical fits demonstrating the distribution of possible ellipses that could fit through these pixels. (b) A 2D visualization of the probability distribution $\mu_a(e)$ over the five-dimensional ellipse parameter space, where contours represent regions of equal probability density and brighter regions indicate higher probability.

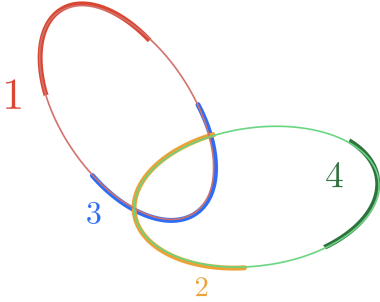


Fig. 3: Illustration of the global view of our approach. The red and blue arcs are correctly grouped as belonging to one ellipse (group 1), while the yellow and green arcs form a second distinct ellipse (group 2). The method successfully identifies compatible arcs based on the global structure in the ellipse parameter space, even when arcs are spatially distant or not directly connected. This example highlights how the co-clustering approach overcomes the limitations of local proximity-based methods by leveraging a comprehensive analysis of arc compatibilities.

While the compatibility matrix may be dense due to the JS divergence computations between all pairs of validated arcs, our method can be extended to handle large-scale applications by leveraging sparse computations. In images with a large number of arcs, the compatibility matrix can be sparsified by selectively retaining compatibility values above a predefined threshold. This optimization reduces memory usage and computational cost while preserving essential compatibility relationships. Since our co-clustering approach relies primarily on the singular vectors of the compatibility matrix rather than its full decomposition, it naturally accommodates sparse Singular Value Decomposition (SVD) techniques. The sparsification process maintains the method’s theoretical guarantees while improving scalability, making it suitable for applications

involving high-resolution images or multiple-image processing where computational efficiency is crucial. This potential for sparse computation provides a practical pathway for handling increasingly complex detection scenarios without compromising detection accuracy.

D. Adaptive Arc Co-Clustering

After computing the compatibilities between arc pairs in the second step, we now proceed to group these arcs into complete ellipses based on the global arc compatibility information. However, existing ellipse detection methods face significant challenges when attempting to perform this grouping effectively. Traditional approaches, such as edge-following methods [10], [11] or Hough Transform-based techniques [16], [17], often struggle with high computational complexity, especially in complex scenes with multiple ellipses. These methods typically rely on exhaustive searches or voting schemes, leading to inefficiencies when dealing with a large number of arc segments [56]. Furthermore, they often require preset thresholds or fixed parameters for determining the number of ellipses, making them inflexible to varying image complexities.

To address these limitations, we propose an adaptive arc co-clustering approach that effectively combines co-clustering algorithms and SVD. Co-clustering, also known as biclustering [39], is a technique that simultaneously clusters rows and columns of a data matrix. This is particularly useful in our context for identifying groups of arc segments that are compatible, regardless of their spatial distances. This capability is essential for detecting ellipses that are large or partially occluded, as it allows for the aggregation of arc information across the image, overcoming the spatial constraints imposed by traditional methods. Moreover, co-clustering adjusts dynamically to the

underlying structure of the data, facilitating more accurate and flexible ellipse formation.

Given the computational complexity of directly applying co-clustering to large matrices, we utilize SVD to reduce this overhead. SVD decomposes the compatibility matrix into three smaller matrices that reveal the most significant underlying patterns. This decomposition identifies the most important relationships between arcs and also significantly reduces the dimensionality of the problem, enhancing processing speed and filtering out less pertinent patterns, which may represent noise. Consequently, the clustering process focuses on the most meaningful data.

By applying co-clustering in the SVD-reduced space, we gain several advantages: a) Improved Efficiency: Working in the reduced space dramatically decreases computational cost. b) Enhanced Accuracy: The combination allows us to group arcs based on their most significant compatibility features, leading to more accurate ellipse detection. c) Flexibility: This approach adapts to different image complexities without requiring preset thresholds or fixed parameters. In the following, we will detail our SVD-based adaptive co-clustering method and demonstrate how it addresses these challenges effectively.

SVD-based Structural Pattern Discovery. After constructing the compatibility matrix $\mathbf{B} \in \mathbb{R}^{n \times n}$ that encapsulates the compatibilities between all detected arcs in Section III-C, we employ SVD to uncover the underlying structural patterns. First, we normalize the compatibility matrix:

$$\tilde{\mathbf{A}} = \mathbf{D}_u^{-\frac{1}{2}} \mathbf{B} \mathbf{D}_v^{-\frac{1}{2}}, \quad (9)$$

where $\mathbf{D}_u \in \mathbb{R}^{n \times n}$ and $\mathbf{D}_v \in \mathbb{R}^{n \times n}$ are diagonal matrices containing the row and column sums of $\mathbf{B}$, respectively. The variable n represents the number of rows and columns in $\mathbf{B}$, corresponding to the number of detected arcs. We then decompose $\tilde{\mathbf{A}}$ to reveal its inherent structure:

$$\tilde{\mathbf{A}} = \mathbf{U} \mathbf{\Sigma} \mathbf{V}^T, \quad (10)$$

where $\mathbf{U} \in \mathbb{R}^{n \times n}$ and $\mathbf{V} \in \mathbb{R}^{n \times n}$ are the left and right singular vectors that capture the fundamental patterns in arc relationships, and $\mathbf{\Sigma} \in \mathbb{R}^{n \times n}$ is a diagonal matrix of singular values indicating the importance of these patterns.

This decomposition is crucial for co-clustering as it reveals the intrinsic structure of arc relationships, separating the row and column patterns of the original compatibility matrix into $\mathbf{U}$ and $\mathbf{V}$, respectively. While this approach also provides computational benefits through dimensionality reduction, its primary advantage lies in uncovering the natural grouping structure that enables effective co-clustering.

Adaptive Determination of Cluster Number. Before proceeding with data projection, we first determine the number of clusters k adaptively. We utilize the properties of SVD to estimate the number of ellipses k using the cumulative energy method [57]:

$$k = \min \left\{ m : \frac{\sum_{i=1}^m \sigma_i^2}{\sum_{i=1}^n \sigma_i^2} \geq \tau \right\}, \quad (11)$$

where σ_i are the singular values from $\mathbf{\Sigma}$, and τ is a predefined threshold, typically set to 0.95 or 0.99. To account for potential underestimations due to noise and occlusions in real-world

scenarios, we scale the estimated k by a factor γ (typically set to 1.4). This scaling helps ensure that we don't miss ellipses when arc segments are fragmented or partially occluded. Our experiments show that this slight overestimation in the number of clusters is more robust than potential underestimation, as the subsequent validation step can effectively filter out spurious detections. More discussion on the choice of τ and γ can be found in Section IV-B.

Data Projection and Co-clustering. After determining k , we project the data onto a new basis by defining $\mathbf{F}$ as:

$$\mathbf{F} = \begin{bmatrix} \mathbf{U}_k \\ \mathbf{V}_k \end{bmatrix}, \quad (12)$$

where $\mathbf{U}_k$ and $\mathbf{V}_k$ are the leading k left and right singular vectors, respectively. This step is the core of our co-clustering approach. Here, n_1 represents the number of rows and n_2 the number of columns in the original compatibility matrix $\mathbf{B}$. The first n_1 rows of $\mathbf{F}$ correspond to the original rows (arcs) of $\mathbf{B}$. The next n_2 rows of $\mathbf{F}$ correspond to the original columns (arcs) of $\mathbf{B}$.

By combining $\mathbf{U}_k$ and $\mathbf{V}_k$, we create a unified space that represents both row and column relationships, enabling simultaneous clustering of both dimensions.

Simultaneous Clustering. We then apply k -means clustering to $\mathbf{F}$, effectively grouping arcs that likely belong to the same ellipse. This step completes the co-clustering process by simultaneously considering the row and column relationships captured in $\mathbf{F}$. Arcs corresponding to rows of $\mathbf{F}$ that are clustered together likely to belong to the same ellipse. The clustering also implicitly groups columns (representing arc compatibilities) that have similar patterns. This simultaneous clustering of rows and columns is the essence of co-clustering. Although we use k -means for the final clustering, the combination of SVD and data projection in $\mathbf{F}$ achieves the effect of co-clustering on the original compatibility matrix.

Remark. In our specific ellipse detection scenario, the compatibility matrix $\mathbf{B}$ is symmetric, meaning that $n_1 = n_2 = n$. Consequently, the singular vectors $\mathbf{U}$ and $\mathbf{V}$ become identical, simplifying the co-clustering process to clustering only the left singular vectors $\mathbf{U}_k$. However, we present the co-clustering algorithm in its general form, allowing for the possibility of non-symmetric matrices where $n_1 \neq n_2$, to demonstrate the broader applicability of the method.

This approach provides a data-driven, adaptive method for determining the number of ellipses, addressing the challenge of varying image complexities. By employing this SVD-based co-clustering framework, we effectively address both the computational complexity issue and the challenge of determining the number of ellipses. This approach allows for efficient and accurate ellipse detection, even in complex images with multiple, potentially overlapping ellipses, significantly advancing the state-of-the-art in ellipse detection methodologies.

E. Implementing Details

In this section, we provide a detailed overview of the implementation of our co-clustering-based ellipse detection method, as shown in Algorithm 1 and the corresponding Function 1.

Algorithm 1 outlines our novel ellipse detection method, which leverages a distribution co-clustering framework to effectively detect ellipses in both synthetic and real-world images. This method addresses key challenges in ellipse detection, including computational complexity, adaptive estimation of the number of ellipses, and the ability to handle complex scenarios such as partial occlusions and overlapping ellipses.

The algorithm begins with the crucial step of arc detection from the input image. This process involves smoothing the image, analyzing gradient information, and generating candidate arc segments. These segments are then merged to form candidate elliptical arcs, which are subsequently validated using the *a contrario* framework [36], [48]. This validation step, represented by the NFA (Number of False Alarms) criterion, ensures that only statistically significant arcs are retained for further processing.

Following arc detection, the algorithm computes a compatibility matrix $\mathbf{B}$, which encapsulates the compatibilities between all validated arcs. This matrix is constructed using JS divergence [20] between arc distributions, providing a probabilistic measure of arc compatibilities. The matrix is then normalized to obtain $\tilde{\mathbf{A}}$, which is subjected to SVD.

The SVD step is crucial as it allows for dimensionality reduction and provides the basis for estimating the number of ellipses in the image. This estimation is performed by the `EstimateNumberOfEllipses` function (Algorithm 1), which uses the cumulative energy of singular values to determine an appropriate number of clusters. This adaptive approach addresses the challenge of varying image complexities.

The algorithm then proceeds to perform co-clustering by applying k -means to the combined matrix of left and right singular vectors ($\mathbf{F}$). This step effectively groups arcs that are likely to belong to the same ellipse. Finally, ellipses are fitted to each cluster and validated, resulting in the set of detected ellipses.

The `EstimateNumberOfEllipses` function, detailed in Algorithm 1, plays a vital role in the overall detection process. It implements the cumulative energy method to estimate the number of ellipses (k) based on the singular values obtained from the SVD. This function first calculates the ratio of cumulative energy up to each singular value to the total energy. It then determines the minimum number of singular values required to reach a predefined threshold τ . To address potential underestimations, the function slightly increases the estimated k by a factor γ .

This algorithmic approach significantly reduces the computational complexity from $O(2^n)$ to $O(nkl)$, where n is the number of arcs, k is the number of clusters, and l is the number of iterations in k -means. Moreover, by operating in the ellipse parameter space rather than image space, the method can effectively handle scenarios with partially occluded ellipses or those with spatially distant arc segments. This global perspective, combined with the probabilistic arc compatibility measure and adaptive cluster estimation, enables robust and accurate ellipse detection across a wide range of image conditions.

Algorithm 1: Ellipse Detection

Input: Image I
Output: Detected Ellipses E
 // Ellipse Detection Algorithm
 1 $A \leftarrow \text{EXTRACTARCS}(I)$; // Extract candidate arcs
 2 $V \leftarrow \emptyset$; // Set of validated arcs
 3 **foreach** $a \in A$ **do**
 4 **if** $\text{NFA}(a) \leq \varepsilon$ **then**
 5 $V \leftarrow V \cup \{a\}$
 6 $\mathbf{B} \leftarrow \text{COMPUTECOMPATIBILITYMATRIX}(V)$
 7 $\mathbf{C} \leftarrow \text{SVD_Co_CLUSTER}(\mathbf{B})$; // Perform co-clustering
 8 $E \leftarrow \emptyset$; // Set of detected ellipses
 9 **foreach** cluster $c \in \mathbf{C}$ **do**
 10 $e \leftarrow \text{FITELLIPSE}(c)$
 11 **if** $\text{VALIDATEELLIPSE}(e)$ **then**
 12 $E \leftarrow E \cup \{e\}$
 13 **return** E

Function 1: SvdCoClusterg

1 **Function** `SVD_Co_Cluster` ($\mathbf{B}$):
 Input: Compatibility matrix $\mathbf{B}$
 Output: Clusters C
 2 $\mathbf{D}_u \leftarrow \text{diag}(\mathbf{B}\mathbf{1})$; // Degree matrix for rows
 3 $\mathbf{D}_v \leftarrow \text{diag}(\mathbf{B}^T\mathbf{1})$; // Degree matrix for columns
 4 $\tilde{\mathbf{A}} \leftarrow \mathbf{D}_u^{-\frac{1}{2}}\mathbf{B}\mathbf{D}_v^{-\frac{1}{2}}$; // Normalize $\mathbf{B}$
 5 $[\mathbf{U}, \Sigma, \mathbf{V}] \leftarrow \text{SVD}(\tilde{\mathbf{A}})$; // Compute SVD
 6 $k \leftarrow \text{ESTIMATENUMBEROFELLIPSES}(\Sigma)$; // Estimate number of clusters
 7 $\mathbf{U}_k \leftarrow \mathbf{U}[:, :k]$; // Take first k left singular vectors
 8 $\mathbf{V}_k \leftarrow \mathbf{V}[:, :k]$; // Take first k right singular vectors
 9 $\mathbf{F} \leftarrow \begin{bmatrix} \mathbf{U}_k \\ \mathbf{V}_k \end{bmatrix}$; // Combine embeddings
 10 $C \leftarrow \text{KMEANS}(\mathbf{F}, k)$; // Cluster combined embeddings
 11 **return** C

Function 2: EstimateNumberOfEllipses

1 **Function** `EstimateNumberOfEllipses` (Σ):
 2 $n \leftarrow$ number of singular values in Σ
 3 $k \leftarrow \min \left\{ m : \sum_{i=1}^m \frac{\sigma_i^2}{\sum_{i=1}^n \sigma_i^2} \geq \tau \right\}$
 4 $k \leftarrow \gamma k$; // Slight increase
 5 **return** k

IV. EXPERIMENTAL RESULTS AND ANALYSIS

This section presents the experimental results of our proposed ellipse detection method. We first describe the datasets and evaluation metrics, followed by a presentation of the quantitative and qualitative results. We then present robustness tests and special scenario analyses. Finally, we discuss the parameters and compare our method with state-of-the-art techniques. All experiments were conducted on a PC equipped with an Intel(R) Core(TM) i9-7900X CPU @ 4.30 GHz, 128 GB of RAM.

A. Experimental Setup

1) *Datasets*: Our evaluation uses both synthetic and real-world datasets to verify the general applicability of the proposed ellipse detector. The code and datasets will be made available online.

For synthetic datasets, we generated images of size 800×600 pixels containing $\beta \in \{6, 9, 12, 15, 18, 21, 24\}$ ellipses as shown in Figure 4. The ellipse parameters were randomly sampled within the following ranges:

- Major axis length: $[50, 150]$ pixels
- Minor axis length: $[30, 100]$ pixels
- Center coordinates: uniformly distributed within image boundaries
- Rotation angle: $[0, 2\pi]$ radians
- At least one pair of ellipses has overlap ratio > 0.3

Salt-and-pepper noise was added using MATLAB's `IMNOISE` function with noise density $d \in \{0.06, 0.12, 0.18, 0.24\}$. Higher d values indicate more severe noise contamination.

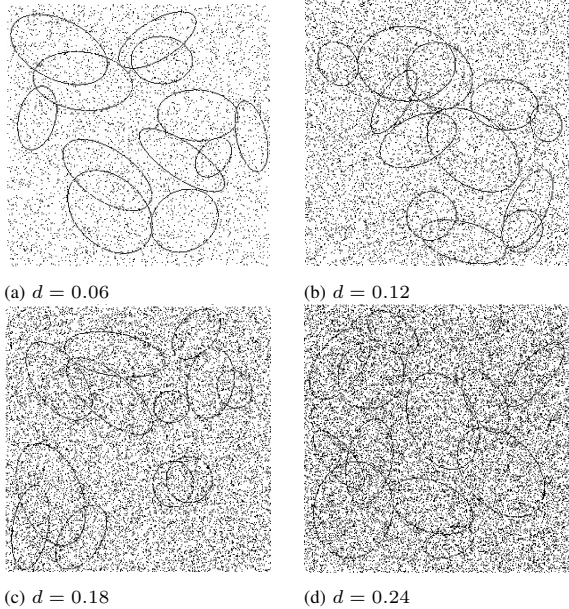


Fig. 4: Examples of synthetic images with different noise densities d . Each image contains 12 ellipses with varying sizes, orientations, and overlap conditions. As d increases, the image becomes increasingly contaminated with salt-and-pepper noise, making ellipse detection more challenging.

For real-world evaluation, we utilized five publicly available datasets:

a) *Prasad++ Dataset*: Only a portion of the original *Prasad* dataset is available online, which was supplemented and relabeled by [14] into the *Prasad+* dataset. However, as illustrated in Figure 5a, the *Prasad+* dataset is incomplete, with some missing ellipses. We further relabeled the *Prasad+* dataset to create a new dataset, *Prasad++*, maintaining the same number of images as the *Prasad+* dataset but with more complete and accurate annotations. Figure 5a and Figure 5b compare the *Prasad+* and *Prasad++* datasets, showcasing the improved ground truth labeling.

b) *Dataset 1*: This dataset comprises 400 images collected from MIRFlickr and LabelMe repositories [34], featuring both high-quality single-object images and noisy low-resolution multi-object scenes.

c) *Dataset 2*: Consisting of 629 images captured by a smartphone camera [34], this dataset presents challenges such as motion blur, varying lighting conditions, and autofocus effects.

d) *Coins Dataset*: We built the *Coins* dataset, consisting of images of coins from different countries and currencies. This dataset challenges ellipse detection methods due to coin overlap, occlusions, and varying lighting conditions, providing a practical test for algorithms in controlled yet complex environments.

e) *Calibration Dataset*: This dataset contains 40 images of calibration patterns [36], featuring multiple ellipses (often more than 70 per image) in concentric structures, providing a challenging scenario for ellipse detection algorithms.

2) *Evaluation Metrics*: We use precision (P), recall (R), and F-measure (F) as our primary metrics. An ellipse is considered correctly detected if its Intersection over Union (IoU) with the ground truth exceeds a threshold θ , typically set to 0.95 for synthetic images [33], [15]. The metrics are calculated as follows:

$$P = \frac{TP}{TP + FP}, R = \frac{TP}{TP + FN}, F = \frac{2P \cdot R}{P + R}, \quad (13)$$

where TP , FP , and FN represent the number of true positives, false positives, and false negatives, respectively.

3) *Baselines*: Our comparative analysis includes several recent state-of-the-art methods published between 2020-2024, ensuring our evaluation encompasses the latest developments in ellipse detection. The methods compared include Zhao *et al.* [15] (2024), Li *et al.* [38] (2022), Shen *et al.* [14] (2021), and Lu *et al.* [13] (2020).

B. Experimental Validation of k Selection

To validate our approach for estimating the number of ellipses (k), we conducted a comprehensive experiment across all datasets, analyzing the relationship between the actual number of ellipses (β) and the estimated k . Fig. 6 presents this relationship as a heatmap, where the x -axis represents β and the y -axis represents k . The heatmap visualizes the frequency of occurrence for each (β, k) pair, with warmer colors indicating higher frequencies. A red reference line ($k = \beta$) is superimposed on the heatmap for comparison.

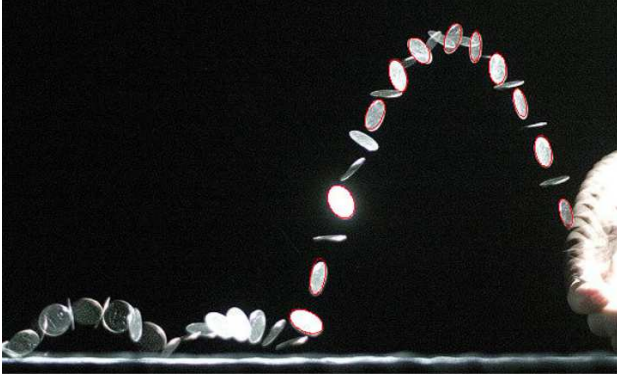
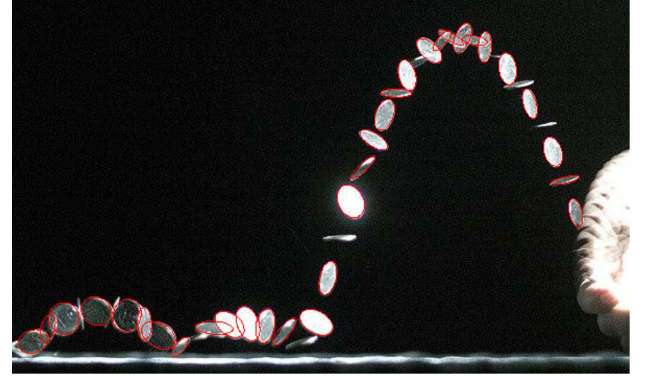
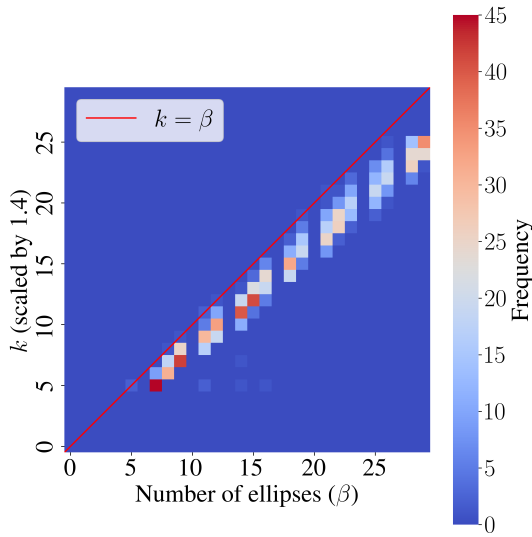
(a) Old *Prasad+* dataset.(b) New *Prasad++* dataset.Fig. 5: Comparison of the *Prasad+* dataset and the *Prasad++* dataset.

Fig. 6: Heatmap showing the relationship between the actual number of ellipses (β) and the estimated number (k). The color intensity indicates the frequency of occurrence for each (β, k) pair. The red line represents $k = \beta$ for reference. The vertical axis shows the scaled number of clusters k ($k = \gamma k$, where $\gamma = 1.4$), which compensates for potential underestimation in complex scenes with noise or occlusions. This scaling factor helps ensure robust detection by allowing for slight overestimation followed by validation.

Analysis of the heatmap in Figure 6 reveals the effectiveness of our k selection method. The red reference line ($k = \beta$) closely hugs the upper boundary of the data point concentration, providing visual confirmation that the condition $k \geq \beta$ holds true for almost all cases. This relationship is crucial, as it demonstrates that our method consistently estimates the number of clusters (k) that is equal to or slightly greater than the actual number of ellipses (β) present in the image.

This slight overestimation strategy proves advantageous in two key ways. Firstly, it ensures sufficient clusters for the co-clustering stage, enabling a thorough exploration of potential ellipse groupings, which is crucial in complex scenarios with partially occluded or overlapping ellipses. Secondly, it maintains overall efficiency by avoiding significant additional computational overhead. Although the method may initially overestimate k , rigorous validation and refitting procedures in the final stages refine these estimates, ensuring accurate

detection of the true number and parameters of ellipses in the image.

C. Evaluation on Synthetic Datasets

Figure 7 presents a comparative analysis of our proposed method against other state-of-the-art techniques on synthetic images under various challenging conditions. The evaluation considers three distinct scenarios: varying numbers of overlapping ellipses, occluded ellipses, and different levels of salt-and-pepper noise.

Figure 7a shows the results for varying numbers of overlapping ellipses, with $\beta \in \{6, 9, 12, 15, 18, 21, 24\}$ representing the number of ellipses in the image. Our method consistently achieves high recall, which translates into superior F-measure performance in all cases. This strong recall performance underscores the effectiveness of our global perspective in detecting true ellipses, even in complex scenarios where ellipses overlap significantly.

Figure 7b illustrates the results for varying numbers of occluded ellipses, where the ellipses are partially hidden. Here, our method demonstrates a more pronounced improvement compared to scenarios with overlapping ellipses. The global view provided by our approach proves particularly advantageous in determining ellipses that are formed by distant arcs with missing segments in between, thereby ensuring high detection accuracy.

Figure 7c presents the performance under different levels of salt-and-pepper noise, with noise density d ranging from 0.06 to 0.24. Our method shows remarkable robustness to noise, achieving the highest F-measure across all noise levels. This robustness is primarily due to our probabilistic model, which effectively manages the uncertainty introduced by noise, resulting in more accurate and reliable ellipse detection.

D. Evaluation on Real-World Datasets

1) *Overall Performance:* The proposed method demonstrated consistently superior performance across all five real-world datasets, achieving the highest F-measure in each case. This consistent top performance underscored the robustness and effectiveness of our approach across various image types and detection scenarios. Table I presents the F-measures,

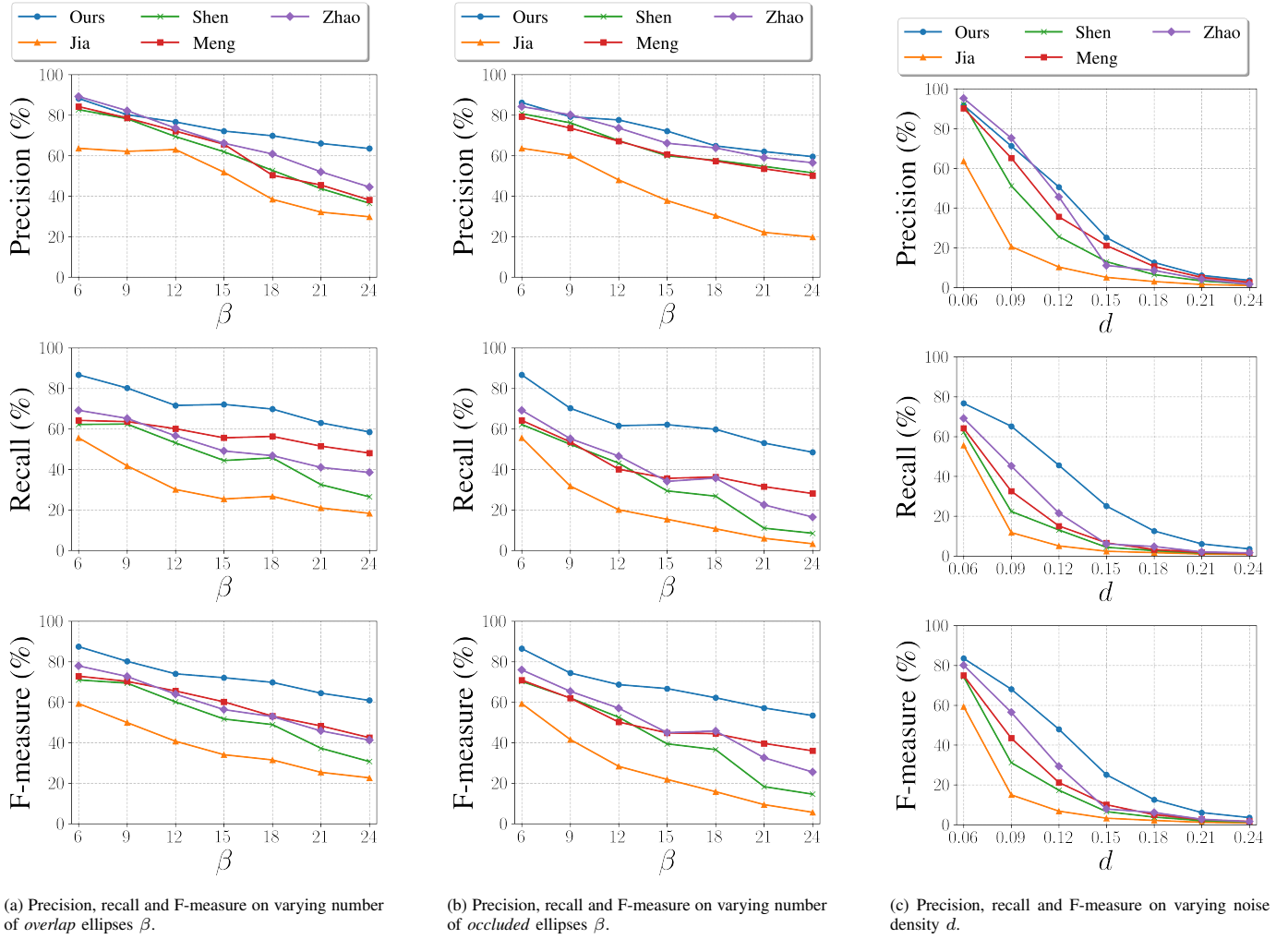


Fig. 7: Comparison of the proposed method with the other methods on the synthetic images.

TABLE I: The F-measures (F), Precision (P), and Recall (R) (%) of the proposed method and methods of [15], [37], [13], [12], [14] on five real-world image datasets. The best results are in **bold**, and the second-best results are underlined.

Method	<i>Prasad++</i>			<i>Dataset 1</i>			<i>Dataset 2</i>			<i>Coins</i>			<i>Calibration</i>		
	F	P	R	F	P	R	F	P	R	F	P	R	F	P	R
[12]	41.10	74.83	28.33	44.23	55.23	36.89	57.12	54.35	60.19	39.70	59.67	29.75	64.08	76.06	55.36
[14]	43.25	75.68	30.28	58.44	68.26	51.09	<u>76.47</u>	76.35	<u>76.6</u>	46.44	65.56	35.96	55.38	92.20	39.58
[37]	46.47	66.3	<u>35.77</u>	59.23	65.4	<u>54.12</u>	67.12	65.35	68.98	54.27	82.33	40.47	77.57	93.59	66.23
[15]	45.95	67.35	34.87	<u>59.47</u>	<u>69.11</u>	52.19	69.02	63.69	75.33	<u>55.83</u>	73.43	45.04	<u>89.69</u>	<u>93.49</u>	<u>86.19</u>
Ours	58.48	69.87	50.28	64.89	70.20	60.33	77.39	<u>71.19</u>	84.77	75.94	<u>79.78</u>	72.46	92.96	93.48	92.44

precision, and recall of our method compared to state-of-the-art techniques on real-world datasets.

In terms of F-measure, our method exhibited a significant advantage. On the *Prasad++* dataset, our approach achieved an F-measure of 58.48%, representing a substantial improvement over the next best method [37], which recorded 46.47%. Similarly, for Dataset 1, we led with an F-measure of 64.89%, outperforming the method [15], which achieved 59.47%. On Dataset 2, our method attained an F-measure of 77.39%, narrowly surpassing the method proposed in [14], which achieved 76.47%. Although the margin here was smaller, our method still prevailed. Notably, on the *Coins* dataset, our approach showed a significant lead, achieving an F-measure of 75.94%,

compared to 55.83% achieved by [15]. On the challenging *Calibration* dataset, our method achieved an F-measure of 92.96%, slightly ahead of the 89.69% achieved by [15].

2) *Precision vs. Recall*: Our method consistently achieved the highest recall across all datasets, indicating a superior ability to detect true ellipses. This strong recall performance was a direct result of the global perspective embedded in our approach, which allowed it to effectively capture and group distant and occluded arcs—an ability that was particularly evident in datasets characterized by high levels of occlusion, such as the *Coins* dataset, where we achieved a recall of 72.46%. Similarly, on the *Calibration* dataset, our recall reached 92.44%, demonstrating our method's effectiveness in

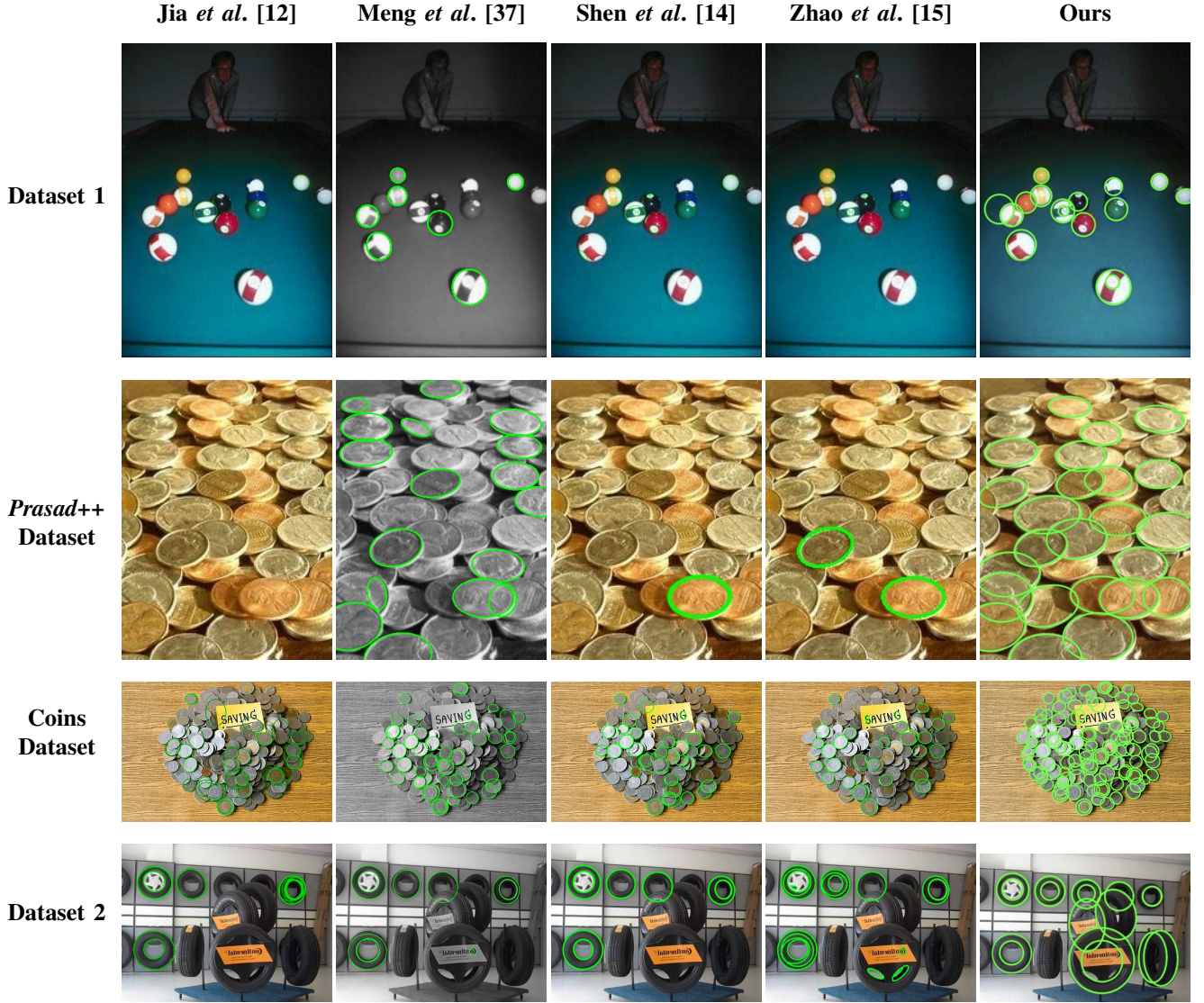


Fig. 8: Qualitative comparison of ellipse detection methods across different datasets. Each row represents a specific dataset with unique challenges: Dataset 1 (reflective surfaces with multiple ellipses), *Prasad++* Dataset (basic ellipse detection), Coins Dataset (overlapping ellipses), and Dataset 2 (industrial application with varying lighting conditions). Columns show results from different methods, demonstrating our method’s superior performance particularly in handling complex scenarios with overlapping and partial ellipses.

scenarios involving multiple, closely spaced ellipses. Although our precision was not always the highest, it remained competitive. For instance, on Dataset 1, our method achieved both the highest precision (70.20%) and recall (60.33%), highlighting the balanced performance of our approach.

3) *Performance on Different Datasets*: The significant improvement observed on the *Coins* dataset, where our method achieved an F-measure of 75.94%, compared to the next best method’s 55.83%, indicated that our approach was particularly effective in handling scenarios involving overlapping or partially occluded ellipses. This performance improvement underscored the advantage of our global view, which facilitated the accurate detection of ellipses under challenging conditions. Similarly, the strong performance on the *Calibration* dataset, where our method achieved an F-measure of 92.96%, further illustrated its capability in accurately detecting multiple, closely spaced ellipses. The consistent high performance

across these diverse datasets demonstrated the versatility and robustness of our approach, making it a highly effective solution for a wide range of ellipse detection challenges in real-world scenarios.

E. Visual Results

Figure 8 presents a qualitative comparison of our proposed method against four state-of-the-art ellipse detection algorithms across various challenging scenarios. The test images include a pool table, coins, many coins, and a tire, each presenting unique challenges for ellipse detection. Our method demonstrates consistently high recall across all test cases, outperforming the compared methods in several aspects.

In the “pool table” scenario, only our method and the method of Meng *et al.* perform well, accurately detecting the elliptical shapes of the balls. The other methods struggle, either failing to capture the elliptical shapes due to the uniform color

or misclassifying partial arcs as noise. Our method’s ability to detect these ellipses with high accuracy is attributed to its global perspective and robust arc grouping process.

The “coins” and “many coins” images present more complex scenarios with multiple ellipses and significant occlusions. In these images, the method of Meng *et al.* demonstrates notable performance, particularly in handling the complexity of the overlapping coins. However, our method shows superior recall, detecting a higher number of ellipses in these highly occluded and cluttered environments. This heightened sensitivity to elliptical shapes in challenging conditions highlights the robustness of our approach.

The “tire” image showcases a particularly challenging case of partial occlusion and spatially distant arc segments belonging to the same ellipse. Our method excels in this scenario, successfully identifying two distant, partially occluded arcs as part of the same ellipse. This capability is a direct result of our algorithm’s global perspective and robust validation process. By correctly associating these distant arcs, our method produces an accurate ellipse detection that passes the validation stage, demonstrating its effectiveness in handling complex, real-world scenarios.

F. Ablation Study

To better understand the effectiveness and efficiency of our proposed method, we conducted comprehensive ablation studies focusing on several key aspects: the choice of divergence measure, parameter sensitivity, and noise robustness.

1) *Comparison of JSD vs. KLD*: We first investigated the impact of using Jensen-Shannon Divergence (JSD) versus Kullback-Leibler Divergence (KLD) for measuring arc compatibility. A key theoretical advantage of JSD over KLD is its symmetry property, which is crucial for unbiased arc compatibility assessment. To quantitatively evaluate the performance difference, we computed the IoU (Intersection over Union) scores for ellipses detected using both methods. Table II presents the detailed comparison across different arc length ranges on the *Prasad++* and *Coins* datasets.

TABLE II: IoU comparison between JSD and KLD based methods. Arc length is measured as the percentage of the full ellipse circumference. Higher IoU indicates better detection accuracy.

Arc Length	Dataset	JSD IoU	KLD IoU
< 15%	Prasad++	0.82	0.61
	Coins	0.79	0.53
15% – 30%	Prasad++	0.87	0.75
	Coins	0.85	0.68
> 30%	Prasad++	0.91	0.88
	Coins	0.89	0.85

The results reveal that KLD’s asymmetric nature particularly affects the detection accuracy of ellipses formed by small arcs (< 15% of full circumference), where the IoU drops significantly compared to JSD (0.82 vs 0.61 on *Prasad++*, 0.79 vs 0.53 on *Coins*). This performance gap narrows for larger arcs but remains notable in the 15% – 30% range. For arcs covering more than 30% of the ellipse circumference, both methods achieve comparable IoU scores, though JSD maintains a slight advantage. The more pronounced difference in the

Coins dataset, which contains numerous partial occlusions and small arc segments, demonstrates that KLD’s asymmetric measure introduces significant bias when evaluating compatibility between arcs of different lengths, particularly affecting the detection of partially visible ellipses in real-world scenarios.

In terms of computational efficiency, both JSD and KLD have the same theoretical complexity. In our implementation, an additional overhead for JSD is negligible compared to the overall processing time of the detection pipeline. Specifically, on our test platform (Intel Core i9-7900X CPU @ 4.30 GHz), the average time difference between JSD and KLD computation per arc pair is approximately 2ms, which represents less than 1% of the total processing time per image.

2) *Parameter Sensitivity Analysis*: We investigated the impact of the threshold parameter τ on both detection performance and computational efficiency. Table III presents the comprehensive evaluation results on the *Prasad++* dataset.

TABLE III: Impact of threshold parameter τ on detection performance and computational efficiency.

τ	F-measure	Precision	Recall	Time (ms)
0.85	46.61%	67.21%	35.67%	87
0.90	52.40%	68.44%	42.45%	113
0.95	58.48%	69.87%	50.28%	245
0.99	58.66%	69.23%	50.89%	598

The experiment results demonstrate a clear trade-off between computational efficiency and detection performance. At lower values ($\tau = 0.85$), the algorithm runs significantly faster (87ms) but achieves poor recall (35.67%), indicating many missed detections despite maintaining reasonable precision. As τ increases to 0.95, both recall and F-measure improve substantially, with processing time increasing to 245ms. Further increasing τ to 0.99 yields minimal improvement in detection metrics (F-measure from 58.48% to 58.66%) while causing a dramatic increase in computational cost (598ms, $2.44\times$ slower than $\tau = 0.95$). The value $\tau = 0.95$ represents an optimal balance point, achieving strong detection performance without the prohibitive computational overhead associated with higher τ values.

3) *Noise Robustness Analysis*: To comprehensively evaluate our method’s robustness against different types of image noise commonly encountered in real-world applications, we conducted extensive experiments with four major types of noise: Salt & Pepper, Gaussian, Speckle, and Poisson. These noise types were chosen to simulate various real-world scenarios: First, Salt & Pepper noise simulates impulse noise from sensor malfunction or transmission errors. Second, Gaussian noise represents thermal noise and sensor limitations. Third, Speckle noise mimics the multiplicative noise common in radar and ultrasound imaging. Finally, Poisson noise simulates the statistical nature of photon detection. For each noise type, we tested three intensity levels (low, medium, high) to evaluate performance degradation patterns. Table IV presents the quantitative results, using F-measure, precision, and recall metrics. The baseline performance on noise-free images (84.81% F-measure, 88.69% precision, 81.25% recall) serves as a reference point.

The results reveal several key findings about our method’s noise robustness:

- **Variable Performance:** The method shows strongest robustness against Salt & Pepper and Gaussian noise, maintaining F-measures above 75% at medium intensity levels.
- **Differential Degradation:** Performance degradation varies by noise type, with Speckle and Poisson noise showing more significant impact even at lower intensities.
- **Critical Thresholds:** Sharp performance drops occur at high noise levels (F-measures ranging from 39.51% to 54.51%), with both precision and recall being substantially affected.
- **Noise Type Sensitivity:** The method shows varying degrees of resilience to different noise types, performing best with Salt & Pepper and Gaussian noise, while showing increased sensitivity to Speckle and Poisson noise.

The observed robustness patterns can be attributed to two key aspects of our method: the global arc compatibility analysis, which helps maintain reliable arc associations under certain noise conditions, and the adaptive co-clustering approach, which can accommodate some noise-induced variations in the compatibility matrix through its dimensionality reduction step.

TABLE IV: Performance evaluation under different noise types. Baseline on noise-free images: 84.81% F-measure, 88.69% precision, 81.25% recall. All Gaussian noises are with mean = 0.

Noise Type	Parameter Setting	F-measure	Precision	Recall
Salt & Pepper	Density = 0.06	82.31%	85.67%	79.23%
	Density = 0.09	78.68%	82.45%	75.21%
	Density = 0.12	52.77%	68.92%	42.78%
Gaussian	$\sigma = 0.05$	81.89%	84.92%	79.05%
	$\sigma = 0.1$	75.69%	78.66%	72.93%
	$\sigma = 0.2$	54.51%	67.33%	45.79%
Speckle	Variance 0.05	76.54%	70.23%	73.12%
	Variance 0.1	51.21%	66.78%	41.67%
	Variance 0.2	39.51%	48.98%	33.12%
Poisson	$\lambda = 0.5$	71.11%	74.12%	68.34%
	$\lambda = 0.2$	67.20%	70.89%	63.89%
	$\lambda = 0.1$	51.45%	67.23%	41.89%

G. Computational Complexity Analysis

To thoroughly evaluate the computational aspects of our method, we conducted a comprehensive analysis examining both theoretical complexity and practical performance metrics. Our analysis encompasses time complexity, space complexity, and actual runtime performance across different test datasets.

1) *Theoretical Complexity:* Let us denote an input image of width W and height H , with n_e edge pixels and n_a detected arcs. The theoretical complexity of our method comprises several sequential computational stages. The initial arc detection and validation process operates at the pixel level, involving convolution operations and gradient computations with a time complexity of $O(W \times H)$ for edge detection. This is followed by arc validation using the *a contrario* framework, which has a complexity of $O(n_e)$.

Following arc detection, the distribution computation stage calculates probability distributions for each arc and computes

pairwise Jensen-Shannon divergences. While individual probability distribution calculations require $O(n_a)$ operations, the pairwise divergence computations lead to an overall complexity of $O(n_a^2)$ for this stage.

The most computationally intensive component is the SVD and co-clustering stage. The SVD computation on the compatibility matrix requires $O(n_a^3)$ operations, and the subsequent k-means clustering adds $O(kn_at)$ operations, where k denotes the number of clusters and t represents the number of iterations. Given that k and t are typically much smaller than n_a , the SVD computation dominates this stage.

The final ellipse fitting and validation phase processes each cluster of arcs with complexity $O(n_c \times m)$, where n_c represents the number of clusters and m denotes the average number of points per cluster. Since both n_c and m are bounded by n_a , this stage does not exceed $O(n_a^2)$.

Therefore, the overall computational complexity of our method is bounded by $O(n_a^3)$, which theoretically exceeds the $O(n_e \log n_e)$ complexity of Jia *et al.* [12] and the $O(n_a^2)$ complexity of Meng *et al.* [37]. This increased theoretical complexity represents the computational cost necessary for our method to capture global arc relationships and achieve superior detection accuracy in challenging scenarios.

2) *Empirical Performance Analysis:* We conducted experiments to measure actual computational costs across different methods. Table V presents the average execution time and memory requirements across our test datasets on our standard testing platform.

TABLE V: Comparison of computational costs across different methods.

Method	Time (ms)	Memory (MB)	Relative Speed
Jia <i>et al.</i> [12]	156	245	1.57×
Meng <i>et al.</i> [37]	189	278	1.30×
Zhao <i>et al.</i> [15]	201	312	1.22×
Shen <i>et al.</i> [14]	178	267	1.38×
Ours	245	356	1.00×

3) *Computational Cost Breakdown:* A detailed analysis of our method’s execution time reveals that:

- Arc detection and validation consumes 15% of the total processing time
- Distribution computation takes 25%, mainly due to pairwise Jensen-Shannon divergence calculations
- SVD and co-clustering stage accounts for 45%, being the most computationally intensive component
- Ellipse fitting and validation uses the remaining 15%

The SVD and co-clustering stage dominates the computational cost, which aligns with our theoretical complexity analysis. This significant computational investment enables our method’s enhanced ability to handle spatially distant arcs and complex geometries effectively. The increased memory usage compared to other methods is primarily due to storing the compatibility matrix and its SVD decomposition, both of which are essential for our global arc compatibility analysis approach.

While our method shows higher computational requirements compared to existing approaches, this overhead enables more robust analysis of challenging cases such as partial occlusions and spatially distant arc segments. The empirical results

demonstrate that our method requires approximately 1.2–1.6 times more computational resources than existing methods, but achieves superior detection accuracy as demonstrated in our previous experimental results.

V. CONCLUSION

In this paper, we have presented a new ellipse detection approach based on a co-clustering procedure to group these candidates, ultimately obtaining the final ellipses. This method revitalizes the global perspective in ellipse detection development, moving away from the reliance on fine-tuning various chaining rules.

Our proposed algorithm has been rigorously tested on both synthetic and real-world images, consistently delivering good results without necessitating manual parameter adjustments across different datasets. The synthetic tests showcase our method's ability to detect small ellipses and its robustness to noise, a consequence of preserving more arc information and leveraging global-level arc grouping. In real-world image tests, our approach consistently achieves the highest F-measures across all four datasets, which further solidifies the algorithm's robustness and adaptability.

By incorporating probabilistic distributions, our method efficiently merges distant yet homologous arcs, addressing the limitations of conventional techniques. The co-clustering procedure successfully handles the high-dimensional features generated by these distributions, resulting in more accurate and efficient ellipse detection.

For future work, we plan to expand the scope of our co-clustering algorithm and explore optimization techniques to improve performance in scenarios with a greater number of correlated arcs and increased complexity. Additionally, to address the longer computation times associated with the high-dimensional distribution calculations, we aim to investigate more efficient approximation algorithms that can reduce computational costs without sacrificing detection accuracy.

ACKNOWLEDGMENT

This work is supported by the Hong Kong Innovation and Technology Commission (InnoHK Project CIMDA), and the Hong Kong Research Grants Council (Project 11204821).

REFERENCES

- [1] Z. Lu, B. Liu, K. Zhang, H. Lin, and Y. Zhang, "A method for measuring the inclination of forgings based on an improved optimization algorithm for fitting ellipses," *IEEE Trans. Instrum. Meas.*, vol. 72, pp. 1–11, 2023.
- [2] I. Ahmed, E. Balestrieri, P. Daponte, R. Imperatore, F. Lamonaca, M. Paolucci, and F. Picariello, "Morphometric measurement of fish blood cell: An image processing and ellipse fitting technique," *IEEE Trans. Instrum. Meas.*, vol. 73, pp. 1–12, 2024.
- [3] M. Li, H. Hu, and H. Yan, "Ice hockey puck tracking through broadcast video," *Neurocomputing*, vol. 551, p. 126484, Sep. 2023.
- [4] H. Dong, J. Zhou, C. Qiu, P. K. Dilip, and I.-M. Chen, "Learning-based ellipse detection for robotic grasps of cylinders and ellipsoids," in *Proc. - IEEE Int. Conf. Robot. Autom.* IEEE, May 2022, pp. 1527–1532.
- [5] J. Muller and K. Dietmayer, "Detecting traffic lights by single shot detection," in *2018 21st Int. Conf. Intell. Transp. Syst. ITSC*, vol. 7. Maui, HI: IEEE, Nov. 2018, pp. 266–273.
- [6] T. Santini, W. Fuhl, and E. Kasneci, "PuRe: Robust pupil detection for real-time pervasive eye tracking," *Comput. Vision Image Understanding*, vol. 170, pp. 40–50, May 2018.
- [7] J. Chen, Y. Zhang, J. Wang, X. Zhou, Y. He, and T. Zhang, "EllipseNet: Anchor-free ellipse detection for automatic cardiac biometrics in fetal echocardiography," in *Med. Image Comput. Comput. Assist. Interv. - MICCAI 2021 - 24th Int. Conf. Strasbg. Fr. Sept. 27 - Oct. 1 2021 Proc. Part VII*, vol. 12907. Springer, 2021, pp. 218–227.
- [8] T. Lefevre, B. Dorizzi, S. Garcia-Salicetti, N. Lemperiere, and S. Belardi, "Effective elliptic fitting for iris normalization," *Comput. Vision Image Understanding*, vol. 117, no. 6, pp. 732–745, 2013.
- [9] Q. Wang and K. L. Boyer, "The active geometric shape model: A new robust deformable shape model and its applications," *Comput. Vision Image Understanding*, vol. 116, no. 12, pp. 1178–1194, 2012.
- [10] E. Kim, M. Haseyama, and H. Kitajima, "Fast and robust ellipse extraction from complicated images," in *Proc. IEEE Int. Conf. Inf. Technol. Appl.*, 2002.
- [11] F. Mai, Y. Hung, H. Zhong, and W. Sze, "A hierarchical approach for fast and robust ellipse extraction," *Pattern Recognit.*, vol. 41, no. 8, pp. 2512–2524, Aug. 2008.
- [12] Q. Jia, X. Fan, Z. Luo, L. Song, and T. Qiu, "A fast ellipse detector using projective invariant pruning," *IEEE Trans. Image Process.*, vol. 26, no. 8, pp. 3665–3679, Aug. 2017.
- [13] C. Lu, S. Xia, M. Shao, and Y. Fu, "Arc-support line segments revisited: An efficient high-quality ellipse detection," *IEEE Trans. Image Process.*, vol. 29, pp. 768–781, 2020.
- [14] Z. Shen, M. Zhao, X. Jia, Y. Liang, L. Fan, and D.-M. Yan, "Combining convex hull and directed graph for fast and accurate ellipse detection," *Graph. Models*, vol. 116, p. 101110, Jul. 2021.
- [15] M. Zhao, X. Jia, L. Ma, L.-M. Hu, and D.-M. Yan, "Coherent chord computation and cross ratio for accurate ellipse detection," *Pattern Recognit.*, vol. 146, p. 109983, Feb. 2024.
- [16] R. O. Duda and P. E. Hart, "Use of the Hough transformation to detect lines and curves in pictures," *Commun. ACM*, vol. 15, no. 1, pp. 11–15, Jan. 1972.
- [17] P. Mukhopadhyay and B. B. Chaudhuri, "A survey of Hough transform," *Pattern Recognit.*, vol. 48, no. 3, pp. 993–1010, Mar. 2015.
- [18] R. A. McLaughlin, "Randomized Hough transform: Improved ellipse detection with comparison," *Pattern Recognit. Lett.*, vol. 19, no. 3–4, pp. 299–305, Mar. 1998.
- [19] R. Halir and J. Flusser, "Numerically stable direct least squares fitting of ellipses," *Proc. Int. Conf. in Central Europe on Computer Graphics, Visualization and Interactive Digital Media*, 1998, 1998.
- [20] J. Lin, "Divergence measures based on the Shannon entropy," *IEEE Transactions on Information Theory*, vol. 37, no. 1, pp. 145–151, 1991.
- [21] G. H. Golub and C. F. Van Loan, *Matrix computations*. JHU press, 2013.
- [22] P. L. Rosin, "Analysing error of fit functions for ellipses," *Pattern Recognit. Lett.*, vol. 17, no. 14, pp. 1461–1470, Dec. 1996.
- [23] A. Fitzgibbon, M. Pilu, and R. Fisher, "Direct least square fitting of ellipses," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 21, no. 5, pp. 476–480, May 1999.
- [24] K. Thurnhofer-Hemsi, E. López-Rubio, E. B. Blázquez-Parra, M. C. Ladrón-de-Guevara-Muñoz, and Ó. D. de-Cózar-Macías, "Ensemble ellipse fitting by spatial median consensus," *Inform. Sci.*, vol. 579, pp. 310–324, Nov. 2021.
- [25] R. Scitovski, K. Sabo, P. Nikić, and S. Majstorović, "A new efficient method for solving the multiple ellipse detection problem," *Expert Syst. Appl.*, vol. 222, p. 119853, Jul. 2023.
- [26] H. W. H. Premachandra, C. Premachandra, C. D. Parape, and H. Kawanaka, "Speed-up ellipse enclosing character detection approach for large-size document images by parallel scanning and Hough transform," *International Journal of Machine Learning and Cybernetics*, vol. 8, no. 1, pp. 371–378, Feb. 2017.
- [27] L. Xu, E. Oja, and P. Kultanen, "A new curve detection method: Randomized Hough transform (RHT)," *Pattern Recognit. Lett.*, vol. 11, no. 5, pp. 331–338, May 1990.
- [28] N. Kiryati, Y. Eldar, and A. Bruckstein, "A probabilistic Hough transform," *Pattern Recognit.*, vol. 24, no. 4, pp. 303–316, 1991.
- [29] Yonghong Xie and Qiang Ji, "A new efficient ellipse detection method," in *Proc. - Int. Conf. Pattern Recognit.*, vol. 2, 2002, pp. 957–960.
- [30] A. Y. S. Chia, M. K. H. Leung, H.-L. Eng, and S. Rahardja, "Ellipse detection with Hough transform in one dimensional parametric space," in *IEEE Int. Conf. Image Process.* San Antonio, TX, USA: IEEE, 2007, pp. 333–336.
- [31] L. Libuda, I. Grothues, and K.-F. Kraiss, "Ellipse detection in digital image data using geometric," in *Proc. First Int. Conf. Comput. Vis. Theory Appl.*, vol. 4. Heidelberg Platz 3, D-14197 Berlin, Germany: Springer Science & Business Media, 2007, p. 229.

- [32] H. Zhou, L. Han, S. Zhu, and H. Yan, "A high-precision ellipse detection method based on quadrant representation and top-down fitting," *Pattern Recognit.*, vol. 154, p. 110603, Oct. 2024.
- [33] M. Fornaciari, R. Cucchiara, and A. Prati, "A mobile vision system for fast and accurate ellipse detection," in *IEEE Conf. Comput. Vis. Pattern Recognit. Workshop*. OR, USA: IEEE, Jun. 2013, pp. 52–53.
- [34] M. Fornaciari, A. Prati, and R. Cucchiara, "A fast and effective ellipse detector for embedded vision applications," *Pattern Recognit.*, vol. 47, no. 11, pp. 3693–3708, Nov. 2014.
- [35] H. Dong, D. K. Prasad, and I.-M. Chen, "Accurate detection of ellipses with false detection control at video rates using a gradient analysis," *Pattern Recognit.*, vol. 81, pp. 112–130, Sep. 2018.
- [36] V. Patraucean, P. Gurdjos, and R. Grompone Von Gioi, "Joint a contrario ellipse and line detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 4, pp. 788–802, Apr. 2017.
- [37] C. Meng, Z. Li, X. Bai, and F. Zhou, "Arc adjacency matrix-based fast ellipse detection," *IEEE Trans. Image Process.*, vol. 29, pp. 4406–4420, 2020.
- [38] F. Li, B. He, G. Li, Z. Wang, and R. Jiang, "Shape-biased ellipse detection network with auxiliary task," *IEEE Trans. Instrum. Meas.*, vol. 71, pp. 1–13, 2022.
- [39] Y. Cheng and G. Church, "Biclustering of expression data," in *Proc. Int. Conf. Intell. Syst. Mol. Biol.*, 2000.
- [40] S. C. Madeira and A. L. Oliveira, "Biclustering algorithms for biological data analysis: A survey," *IEEE/acm Trans. Comput. Biol. Bioinformatics*, 2004.
- [41] F. Angiulli, E. Cesario, and C. Pizzuti, "Random walk biclustering for microarray data," *Inf. Sci.*, vol. 178, no. 6, pp. 1479–1497, Mar. 2008.
- [42] S. Daruru, N. M. Marin, M. Walker, and J. Ghosh, "Pervasive parallelism in data mining: Dataflow solution to co-clustering large and sparse netflix data," in *Proc. 15th ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.* New York, NY, USA: Association for Computing Machinery, Jun. 2009, pp. 1115–1124.
- [43] N. Liu, F. Chen, and M. Lu, "Spectral co-clustering documents and words using fuzzy k-harmonic means," *International Journal of Machine Learning and Cybernetics*, vol. 4, no. 1, pp. 75–83, Feb. 2013.
- [44] S. Khan, L. Chen, and H. Yan, "Co-clustering to reveal salient facial features for expression recognition," *IEEE Trans. Affective Comput.*, vol. 11, no. 2, pp. 348–360, Apr. 2020.
- [45] Z. Wei, H. Zhao, L. Zhao, and H. Yan, "Multiscale co-clustering for tensor data based on canonical polyadic decomposition and slice-wise factorization," *Inf. Sci.*, vol. 503, pp. 72–91, Nov. 2019.
- [46] J. Wei, H. Ma, Y. Liu, Z. Li, and N. Li, "Hierarchical high-order co-clustering algorithm by maximizing modularity," *International Journal of Machine Learning and Cybernetics*, vol. 12, no. 10, pp. 2887–2898, Oct. 2021.
- [47] J. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability/University of California Press*, 1967.
- [48] V. Pătraucean, P. Gurdjos, and R. G. Von Gioi, "A parameterless line segment and elliptical arc detector with enhanced ellipse fitting," in *Computer Vision—ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, October 7–13, 2012, Proceedings, Part II 12*. Springer, 2012, pp. 572–585.
- [49] D. H. Marimont and Y. Rubner, "A probabilistic framework for edge detection and scale selection," in *Sixth International Conference on Computer Vision (IEEE Cat. No. 98CH36271)*. IEEE, 1998, pp. 207–214.
- [50] M. Prandini, J. Hu, J. Lygeros, and S. Sastry, "A probabilistic approach to aircraft conflict detection," *IEEE Transactions on intelligent transportation systems*, vol. 1, no. 4, pp. 199–220, 2000.
- [51] A. Lejeune, J. G. Verly, and M. Van Droogenbroeck, "Probabilistic framework for the characterization of surfaces and edges in range images, with application to edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 9, pp. 2209–2222, 2017.
- [52] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, 2nd ed. John Wiley & Sons, 2012.
- [53] J. M. Bernardo and A. F. Smith, *Bayesian theory*. John Wiley & Sons, 2009, vol. 405.
- [54] G. G. Simpson, "Notes on the measurement of faunal resemblance," *American Journal of Science*, vol. 258, no. A, pp. 300–311, 1960.
- [55] S. Kullback and R. A. Leibler, "On information and sufficiency," *The Annals of Mathematical Statistics*, vol. 22, no. 1, pp. 79–86, 1951.
- [56] X. Zhang, D. Liu, X. Wang, and X. Zhang, "Advanced ellipse fitting algorithm based on ADMM and hybrid BFGS method," *IEEE Trans. Instrum. Meas.*, vol. 70, pp. 1–11, 2021.
- [57] I. T. Jolliffe, *Principal Component Analysis*, 2nd ed., ser. Springer Series in Statistics. Springer, 2002.



Zihan Wu received a Bachelor's degree in Physics and a second Bachelor's degree in Mathematics and Applied Mathematics from the School for the Gifted Young at the University of Science and Technology of China (USTC) in June 2020. He is currently pursuing a Ph.D. in Electrical Engineering at the City University of Hong Kong. His research interests include data mining, pattern recognition, and natural language processing.



Zhaoke Huang received the PhD degree in control science and engineering from Central South University, Changsha, China, in 2019. He is currently a Postdoctoral Fellow with the Department of Electrical Engineering, City University of Hong Kong. His current research interests include computer vision and bioinformatics. He was a recipient of the Outstanding Doctoral Dissertation Award of the Nonferrous Metals Society of China (NFSOC) in 2020, and Hunan Postdoctoral Program for Innovative Talent in 2020.



Hong Yan received his PhD degree from Yale University. He was Professor of Imaging Science at the University of Sydney and currently is Wong Chun Hong Professor of Data Engineering and Chair Professor of Computer Engineering at City University of Hong Kong. Professor Yan's research interests include image processing, pattern recognition, and bioinformatics. He has over 600 journal and conference publications in these areas.

Professor Yan is an IEEE Fellow and IAPR Fellow. He received the 2016 Norbert Wiener Award from the IEEE SMC Society for contributions to image and biomolecular pattern recognition techniques. He is a member of the European Academy of Sciences and Arts and a Fellow of the US National Academy of Inventors.