

# 從像素到感知：幾何特徵、相機視角挑戰與現代視覺演算法綜合報告

## 第一部分：電腦視覺中幾何特徵的基礎

### 1.1 定義「抽象」幾何特徵

在電腦視覺領域，幾何特徵堪稱是解讀視覺世界的「字母」。它們是從複雜的圖像數據中提取出的簡化而富有意義的表徵，其核心目標是將數以百萬計的像素點，轉化為一組稀疏、結構化且易於分析的資訊<sup>1</sup>。這個抽象化的過程是實現物體辨識、分類、三維重建等高階視覺任務的基石<sup>1</sup>。

這個抽象過程並非隨意為之，而是一種深思熟慮的權衡——刻意捨棄部分資訊以換取更重要的特性。傳統的特徵提取方法，其智慧之處不僅在於處理所有可見資訊，更在於判斷哪些資訊可以被安全地忽略。例如，演算法會忽略一堵白牆上大面積的均勻色塊，而將注意力集中在牆角、邊緣等光影和結構急劇變化的位置<sup>3</sup>。這種做法的背後邏輯是，一個角落的幾何位置（由幾個座標點定義）遠比構成它的成千上萬個像素點更為穩定。當相機視角輕微移動或光照條件發生變化時，像素的顏色和亮度值可能劇烈波動，但「角落」這個幾何概念本身卻依然存在。因此，「抽象」的本質是一種為了獲取魯棒性（robustness）而進行的策略性資訊捨棄。它將視覺世界從像素級的繁雜，提煉為幾何層面的精簡，從而讓機器能夠在多變的環境中做出更穩定的判斷。

幾何特徵主要可分為以下幾種類型：

- **原始特徵 (Primitive Features)**: 這些是構成視覺資訊的基本單元。
  - **點 (Points)**: 也稱為角點 (corners) 或關鍵點 (keypoints)，是零維度的特徵，代表圖像中特定且顯著的位置。這些點通常是圖像梯度在多個方向上都發生劇烈變化的區域，因此具有很高的穩定性和可重複性，不易因視角或光照的微小變化而消失<sup>3</sup>。典型的點特徵包括由 Harris 或 FAST 等演算法偵測出的角點，以及由 SIFT 演算法偵測出的斑點 (blobs) 或顯著點 (salient points)<sup>3</sup>。
  - **線 (Lines)**: 也稱為邊緣 (edges) 或脊線 (ridges)，是一維的特徵，通常表示不同圖像區域之間的邊界或物體的對稱軸<sup>4</sup>。線特徵對於描述物體的輪廓和結構至關重要<sup>1</sup>。Canny、Sobel 等邊緣偵測演算法被廣泛用於提取這類特徵<sup>1</sup>。
  - **區域 (Regions)**: 也稱為斑點 (blobs) 或紋理 (textures)，是二維的特徵，代表了具有共同屬性（如均勻的顏色、強度或紋理）的圖像區域<sup>4</sup>。斑點偵測是識別這類區域的關鍵技術<sup>5</sup>。
- **複合與高階特徵 (Compound and Higher-Level Features)**: 原始特徵可以組合起來，形成更複雜、更具描述性的複合特徵。這包括由多個原始特徵（如邊緣和角點）組成的幾何組

件特徵，以及透過布林運算(例如交集或聯集)結合的布林複合特徵<sup>5</sup>。這一概念為後續章節將要探討的、更為先進的「物體級特徵」奠定了概念基礎。

總而言之，特徵提取的最終目標是從圖像中找出那些具有獨特性 (distinctive) 和可重複性 (repeatable) 的特徵，並且這些特徵必須對視角、尺度和光照的變化具有強大的魯棒性<sup>1</sup>。透過這種方式，電腦視覺系統得以簡化數據、提高後續任務的準確性、增強對噪點的抵抗力，並最終實現即時處理與決策<sup>1</sup>。

## 1.2 手工設計特徵演算法分類：偵測-描述-匹配流程

在深度學習興起之前，電腦視覺領域長期由「手工設計特徵」(Hand-crafted Features) 的方法主導。這些方法遵循一個經典的三步流程：偵測 (Detect)、描述 (Describe) 和匹配 (Match)<sup>8</sup>。

1. 特徵偵測：在圖像中尋找穩定且可重複的關鍵點。
2. 特徵描述：為每個偵測到的關鍵點生成一個獨特的「簽名」(即描述子)，這個簽名需要對各種變換保持不變性。
3. 特徵匹配：比較兩幅圖像中的特徵描述子，找到對應的特徵點對。

以下將深入探討三種最具代表性的手工設計特徵演算法：SIFT、SURF 和 ORB。

### SIFT (Scale-Invariant Feature Transform)

SIFT 是由 David Lowe 提出的，長久以來被視為特徵匹配領域的黃金標準。其核心目標是找到對尺度、旋轉不變，且對光照和反射變換具有魯棒性的特徵點<sup>10</sup>。它透過在圖像的尺度空間中尋找穩定的極值點來實現這一目標<sup>3</sup>。其流程分為四個主要步驟：

1. 尺度空間極值偵測 (Scale-Space Extrema Detection)：首先，透過將原始圖像與不同標準差 ( $\sigma$ ) 的高斯濾波器進行卷積，建立一個高斯金字塔。接著，將金字塔中相鄰的高斯模糊圖像相減，得到一系列高斯差分 (Difference-of-Gaussians, DoG) 圖像。DoG 是高斯-拉普拉斯 (Laplacian of Gaussian, LoG) 算子的一種高效近似，能夠有效地偵測圖像中的斑點狀特徵。演算法會在 DoG 空間中(跨越圖像位置和尺度)尋找局部極值點(最大值或最小值)，這些點即為候選關鍵點<sup>3</sup>。
2. 關鍵點定位 (Keypoint Localization)：對找到的候選關鍵點進行精確的定位，達到子像素級別的精度。同時，演算法會過濾掉那些對比度過低或位於不穩定邊緣上的點。例如，一條直線的邊緣在垂直於邊緣的方向上響應很強，但在平行方向上響應很弱，這類點容易受噪點影響，因此會被剔除。這個步驟極大地增強了特徵的穩定性和抗噪能力<sup>3</sup>。
3. 方向賦值 (Orientation Assignment)：為了實現旋轉不變性，需要為每個關鍵點分配一個標準方向。演算法會在關鍵點周圍的鄰域內計算梯度的大小和方向，並建立一個梯度方向直方圖。直方圖的峰值所對應的方向就被指定為該關鍵點的主方向。如果存在多個顯著的峰值，也可以為同一個關鍵點分配多個方向，這有助於提高匹配的魯棒性<sup>3</sup>。
4. 描述子生成 (Descriptor Generation)：最後，在每個關鍵點周圍選取一個 16x16 像素的鄰域。將這個鄰域旋轉至其主方向，以確保描述子的旋轉不變性。接著，將該鄰域劃分為一個 4x4 的網格，每個子區域為 4x4 像素。在每個子區域內，計算一個 8 個方向的梯度直方圖。最後，將這 16 個子區域的直方圖串聯起來，形成一個 4x4x8=128 維的特徵描述子向量。這

個高維度的向量就是該關鍵點的獨特「指紋」<sup>12</sup>。

## SURF (Speeded-Up Robust Features)

SURF 旨在成為 SIFT 的一種更快速的替代方案，力求在保持相當性能的同時大幅提升計算效率<sup>6</sup>。其效率的提升主要來自兩項關鍵創新：

1. 積分圖像 (Integral Images): SURF 引入了積分圖像這一數據結構。透過積分圖像，計算圖像中任意矩形區域內像素值的和只需要幾次加減法運算。這使得 SURF 可以使用盒式濾波器 (Box Filter) 來近似高斯濾波器，極大地加速了卷積過程，避免了 SIFT 中耗時的高斯模糊操作<sup>12</sup>。
2. 基於 Hessian 矩陣的偵測器: SURF 使用基於 Hessian 矩陣行列式的斑點偵測器來尋找特徵點。同樣地，藉助積分圖像，Hessian 矩陣的計算可以被高效完成<sup>12</sup>。

在描述子方面，SURF 計算關鍵點周圍的 Haar 小波響應，並生成一個 64 維的描述子向量。這個描述子不僅維度更低（有利於加速匹配），計算速度也比 SIFT 的描述子更快<sup>12</sup>。

## ORB (Oriented FAST and Rotated BRIEF)

ORB 是一種速度極快且無專利限制的特徵提取演算法，特別適用於即時應用場景，例如在移動設備上進行特徵匹配<sup>17</sup>。ORB 是對兩種現有技術的巧妙融合與改進：

1. 偵測器 (FAST): ORB 使用 FAST (Features from Accelerated Segment Test) 角點偵測器。FAST 透過檢查候選點周圍圓環上的一圈像素點的強度，快速判斷該點是否為角點，其計算效率遠高於 SIFT 和 SURF 的偵測器<sup>6</sup>。ORB 進一步使用 Harris 角點響應值對 FAST 偵測到的角點進行排序和過濾，以選取最優質的特徵點<sup>20</sup>。
2. 描述子 (BRIEF): ORB 採用 BRIEF (Binary Robust Independent Elementary Features) 描述子。BRIEF 透過在關鍵點鄰域內比較一系列隨機點對的像素強度，生成一個二進位字串作為描述子。例如，如果點 A 的強度大於點 B，則該位為 1，否則為 0。這種二進位描述子的計算速度極快，並且在匹配時可以使用漢明距離 (Hamming Distance) 進行比較，這比計算 SIFT/SURF 浮點數向量之間的歐幾里得距離要快得多<sup>6</sup>。
3. 旋轉不變性 (ORB 的 "O" 和 "R"): 原始的 BRIEF 描述子不具備旋轉不變性。ORB 的核心創新在於解決了這個問題。它採用強度質心 (intensity centroid) 的方法來確定每個關鍵點的主方向。具體來說，演算法計算關鍵點鄰域圖像塊的矩，並找到其質心位置。從圖像塊的幾何中心指向質心的向量便定義了該關鍵點的方向<sup>17</sup>。在生成 BRIEF 描述子之前，ORB 會根據這個計算出的主方向，旋轉採樣點對的模式，從而使得最終生成的二進位描述子具有了旋轉不變性，這就是所謂的「Steered BRIEF」<sup>19</sup>。

## 經典特徵演算法比較分析

為了在實際應用中做出明智的選擇，下表對 SIFT、SURF 和 ORB 的關鍵特性進行了綜合比較。

特性	SIFT (尺度不變特徵轉換)	SURF (加速穩健特徵)	ORB (定向 FAST 與旋轉 BRIEF)
核心原理	在高斯差分尺度空間中偵測極值點，並計算梯度方向直方圖描述子 <sup>3</sup> 。	使用積分圖像加速 Hessian 矩陣的計算來偵測特徵點，並計算 Haar 小波響應描述子 <sup>16</sup> 。	結合 FAST 角點偵測器和 BRIEF 二進位描述子，並透過強度質心法實現旋轉不變性 <sup>6</sup> 。
偵測器	高斯差分 (DoG)	快速 Hessian 矩陣	FAST 角點偵測器 (經 Harris 評分過濾)
描述子類型	浮點數向量	浮點數向量	二進位字串
描述子維度	128 維 <sup>16</sup>	64 綴 <sup>16</sup>	256 位 (32 字節) <sup>16</sup>
匹配度量	歐幾里得距離	歐幾里得距離	漢明距離
相對速度	較慢 (基準) <sup>16</sup>	中等 (約比 SIFT 快 3 倍) <sup>13</sup>	最快 (比 SIFT 快約 100 倍) <sup>8</sup>
尺度不變性	非常好 <sup>16</sup>	好 <sup>16</sup>	有限，依賴圖像金字塔，對大尺度變化較不穩定 <sup>16</sup> 。
旋轉不變性	非常好 <sup>16</sup>	好 <sup>12</sup>	好，但對大角度旋轉的魯棒性略遜於 SIFT <sup>16</sup> 。
魯棒性	對光照、噪點、仿射變換等具有很強的魯棒性 <sup>10</sup> 。	對光照和模糊有較好的魯棒性，但對視角變化不如 SIFT <sup>12</sup> 。	對光照變化和噪點較敏感，適用於變化較小的場景 <sup>6</sup> 。
專利狀態	曾受專利保護，現已過期。	曾受專利保護，現已過期。	完全開源，無專利限制 <sup>17</sup> 。

這張比較表不僅僅是事實的羅列，更是一個實用的決策指南。對於一個開發者而言，選擇哪種演算法取決於應用的具體需求。如果應用場景（如離線三維重建）追求最高的精度和魯棒性，且可以承擔較高的計算成本，那麼 SIFT 仍然是一個強有力的選擇<sup>13</sup>。反之，如果應用需要部署在計算資源有限的移動設備上，並且要求即時響應，那麼 ORB 因其驚人的速度和免費的特性而成為不二之選<sup>17</sup>。SURF 則提供了一個介於兩者之間的平衡選擇。

## 第二部分：相機視角下的挑戰——不變性與魯棒性

### 2.1 不變性原理：連接三維世界與二維圖像的橋樑

相機的本質是將三維物理世界投影到二維的圖像平面上。在這個過程中，同一個三維物體上的點，會因為相機的位姿（位置和方向）、與物體的距離以及環境光照條件的不同，而在二維圖像上呈現出截然不同的外觀<sup>23</sup>。這就引出了電腦視覺中的一個核心問題：如何讓演算法認識到，兩幅圖像中兩個看起來不盡相同的圖像塊，實際上對應的是同一個物理點？

答案就在於不變性 (Invariance)。一個特徵如果對於某種變換是「不變的」，意味著即使圖像經歷了該變換，該特徵的描述子仍然保持不變或極其相似<sup>23</sup>。不變性是實現特徵匹配的根本，是讓機器能夠跨越視角、尺度和光照的鴻溝，穩定地理解和識別現實世界物體的關鍵<sup>10</sup>。

特徵不變性主要可以分為兩大類：

- 幾何不變性 (Geometric Invariance): 應對物體在空間中的幾何形態變化。
  - 平移 (Translation): 特徵本身是局部的，因此天然地對平移具有不變性。
  - 旋轉 (Rotation): 物體在圖像平面內的旋轉<sup>10</sup>。
  - 尺度 (Scale): 物體因遠近變化導致在圖像中尺寸的改變<sup>10</sup>。
  - 仿射 (Affine): 更複雜的變換，包括剪切和非均勻縮放，通常由劇烈的視角變化引起（例如，從斜側面觀察一個平面物體）<sup>10</sup>。
- 光度不變性 (Photometric Invariance): 應對光照條件的變化。
  - 光照 (Illumination): 圖像的整體亮度、對比度的變化<sup>10</sup>。

## 2.2 實現不變性的技術剖析

為了應對上述挑戰，研究人員開發了多種精巧的技術來賦予特徵所需的不變性。

- 透過尺度空間實現尺度不變性
  - 理論核心：「尺度空間」(Scale-Space) 理論是實現尺度不變性的基石。其核心思想是，透過將圖像與一系列尺度遞增的高斯核進行卷積（即進行不同程度的模糊），來模擬物體在不同遠近距離下的外觀<sup>3</sup>。一個真正穩定的特徵，應該在某個特徵尺度上達到響應的極值，並且這個極值在空間和尺度兩個維度上都是局部的<sup>10</sup>。
  - SIFT 的 DoG 實現：如前所述，SIFT 使用高斯差分 (DoG) 金字塔作為尺度空間的高效近似。一個特徵點之所以被認為是尺度不變的，是因為它在 DoG 金字塔的多個層級上都表現為一個穩定的極值點。演算法透過這種方式，不僅偵測到了特徵點，還同時確定了該特徵點的「特徵尺度」(characteristic scale)，即該特徵最顯著的尺寸<sup>3</sup>。
- 透過方向正規化實現旋轉不變性
  - 核心策略：通用的解決方案是為每個特徵點估算出一個標準的、唯一的方向，然後在計算描述子時，將特徵鄰域旋轉到這個標準方向上，從而消除旋轉帶來的影響<sup>14</sup>。
  - SIFT 的梯度直方圖：SIFT 在特徵點鄰域內統計梯度的方向分佈，並將分佈的峰值作為主方向。這種方法非常魯棒，但計算量較大<sup>3</sup>。
  - ORB 的強度質心：ORB 則採用一種更為高效的方法。它計算特徵點鄰域內像素強度的「質心」（或稱「重心」）。從鄰域的幾何中心指向這個強度質心的向量，便提供了一個快速且穩定的方向估計<sup>17</sup>。這種方法的計算成本遠低於 SIFT 的直方圖方法<sup>17</sup>。
- 實現光照與仿射不變性
  - 光照不變性：主要透過描述子正規化來實現。例如，SIFT 在生成 128 維描述子向量後，會將其正規化為單位長度。這一步操作可以有效地抵消圖像整體亮度的線性變化。此外，在計算梯度時對幅值設定一個上限，可以增強對比度變化的魯棒性<sup>10</sup>。
  - 仿射不變性：實現完全的仿射不變性更具挑戰性。雖然 SIFT 和 SURF 對於輕微的仿射變換具有一定的魯棒性，但當視角變化劇烈時，它們的性能會下降。專門的仿射不變偵測器，如 MSER (Maximally Stable Extremal Regions)，能夠偵測出在不同強度

閾值下形狀保持高度穩定的區域，這些區域對仿射變換具有很強的魯棒性<sup>23</sup>。另一種方法是，在偵測到特徵點後，擬合一個橢圓來描述其局部形狀，並利用該橢圓的幾何屬性（如主軸方向和軸長比）來定義一個仿射正規化的坐標系，然後再計算描述子<sup>23</sup>。

## 2.3 克服真實世界中的匹配挑戰

除了上述理想化的幾何與光度變換，從相機視角出發，特徵匹配還會面臨更多來自真實世界的複雜挑戰。

- 真實相機視角的挑戰
  - 視角變化 (Viewpoint Change): 當相機從不同角度拍攝同一物體時，會產生明顯的透視和仿射畸變，這對描述子的穩定性構成了嚴峻考驗<sup>26</sup>。
  - 遮擋 (Occlusion): 物體的一部分可能被其他物體遮擋，導致特徵點缺失或特徵描述不完整（例如，一條完整的直線被柱子遮擋後，在圖像中變成了兩條不連續的線段），從而導致匹配失敗<sup>28</sup>。
  - 雜亂背景與重複紋理 (Clutter and Repetitive Patterns): 雜亂的背景會引入大量無關的特徵點，增加了錯誤匹配的概率。而重複的紋理（如磚牆、棋盤格）則會產生歧義，因為許多不同的特徵點可能擁有幾乎完全相同的描述子，使得演算法難以區分<sup>31</sup>。
- 內點/外點問題與 RANSAC 演算法
  - 問題的提出：由於上述挑戰的存在，任何特徵匹配演算法產生的初始匹配對集合中，都不可避免地會包含一部分正確的匹配（稱為內點/inliers）和一部分錯誤的匹配（稱為外點/outliers）<sup>33</sup>。如果直接使用這個混雜的集合來計算幾何變換（例如，用於圖像拼接的單應性矩陣），那麼外點會嚴重扭曲計算結果，導致完全錯誤的估計<sup>33</sup>。
  - RANSAC (Random Sample Consensus) 作為解決方案：RANSAC 是一種迭代式的、基於投票的魯棒估計方法，其核心思想是從充滿噪聲的數據中分離出內點，並基於這些內點來估計模型參數<sup>33</sup>。
  - 直觀解釋：可以將 RANSAC 的過程類比為在一張散佈著許多點的紙上尋找一條直線，其中大部分點大致在線上，但有幾個點是隨機的「噪點」。傳統的最小二乘法會試圖遷就所有點，結果得到的直線會被噪點拉偏。而 RANSAC 的做法是：隨機選取最少量的點（對直線而言是 2 個點）來定義一個候選模型（一條直線），然後去「驗證」這個模型——數一數數據集中有多少其他的點離這條線足夠近。這個過程重複多次，每次都用不同的隨機點集來生成候選模型。最終，那條獲得「票數」最多（即擁有最多內點支持）的直線，就被認為是最佳模型。那些不符合最佳模型的點，則被歸類為外點<sup>36</sup>。
  - 在特徵匹配中的正式步驟：
    1. 隨機採樣：從所有匹配對中，隨機選取一個最小子集（例如，計算單應性矩陣需要 4 對匹配點）。
    2. 計算模型：僅使用這個最小子集來計算一個候選的幾何模型（例如，單應性矩陣 H）。
    3. 驗證與評分：將數據集中所有其他的匹配對，用這個計算出的模型 H 進行驗證。如果一對匹配點經過 H 變換後的誤差（如重投影誤差）小於某個預設的閾值，則將其歸類為內點。統計該模型獲得的內點總數。
    4. 迭代：重複上述 1-3 步，直到達到預設的迭代次數 k。

5. 選取最佳模型：選擇在所有迭代中，獲得內點數量最多的那個模型作為最終的穩健估計結果。
  6. (可選) 模型優化：使用上一步選出的最佳模型所對應的所有內點，重新計算一次模型參數，以獲得更高的精度。
- RANSAC 的重要性：RANSAC 是電腦視覺中許多應用（如圖像拼接、三維重建、視訊穩定）的基石。它提供了關鍵的數據清洗步驟，使得這些應用能夠在充滿噪聲和錯誤的真實世界數據上穩健地工作<sup>34</sup>。

RANSAC 演算法的成功揭示了一個關於我們所處物理世界的基本假設：結構並非隨機。該演算法之所以有效，是因為它相信，對應於真實世界中某個潛在幾何結構（例如，場景中的一個平面）的內點，會持續、一致地「投票」給同一個正確的數學模型（例如，正確的單應性矩陣）。而外點，作為本質上的隨機誤差，其投票行為是分散的，不會集中支持任何一個特定的模型<sup>33</sup>。因此，RANSAC 的成功建立在一個前提之上：場景可以被分解為由簡單幾何實體主導的結構。這也暴露了它的潛在弱點：如果場景本身缺乏一個主導性的、簡單的幾何結構（例如，拍攝一張揉皺的紙，或一個複雜的非剛性物體），RANSAC 可能會失敗，因為沒有任何一個模型能夠獲得足夠多的內點支持。這也解釋了為什麼對非剛性物體進行匹配是一個遠為困難的問題，通常需要截然不同的技術來解決<sup>26</sup>。

## 第三部分：現代紀元——基於學習的特徵提取

### 3.1 範式轉移：深度學習的興起

儘管像 SIFT 這樣的手工設計特徵功能強大，但它們在某些極具挑戰性的場景中會達到性能瓶頸。例如，在紋理稀疏的區域（如白牆）、光照變化極端（如白天與黑夜的對比）或天氣條件惡劣的環境下，手工特徵的性能會嚴重下降，因為它們所依賴的梯度統計等底層假設不再成立<sup>31</sup>。這些演算法的設計基於人類工程師的先驗知識和幾何直覺，未必是解決匹配問題的最優解<sup>41</sup>。深度學習為此提供了一種全新的、數據驅動的解決方案。與其由人類預先定義「什麼是好的特徵」，不如讓神經網路直接從海量數據中學習到最適合匹配任務的特徵表示<sup>39</sup>。實踐證明，卷積神經網路（CNN）學習到的特徵表示，在魯棒性上通常優於手工設計的表示<sup>39</sup>。

### 3.2 學習特徵的演進之路

基於學習的特徵提取方法也經歷了一個逐步演進的過程。

- 第一階段：學習更優的描述子
 

早期的研究保留了傳統的特徵偵測器（如 SIFT 偵測器），但用一個深度學習模型來取代手工設計的描述子生成步驟。

  - 核心方法：通常使用孿生網路（Siamese Network）或三元組網路（Triplet Network）結構，對從圖像中提取出的圖像塊（patch）進行訓練。其目標是學習一個嵌入空間（embedding space），在這個空間中，來自同一物理點的匹配圖像塊之間的距離很近

, 而來自不同點的非匹配圖像塊之間的距離則很遠<sup>42</sup>。

- 代表模型: MatchNet<sup>43</sup>、DeepDesc<sup>43</sup>、TFeat<sup>43</sup>、L2-Net<sup>43</sup> 和 HardNet<sup>43</sup> 等模型, 透過不斷改進網路架構和損失函數(例如, 使用帶有難例挖掘(hard negative mining)的三元組損失), 逐步提升了描述子的獨特性和魯棒性。
- 第二階段: 學習完整的端到端流程
  - 可微分性的挑戰: 將整個偵測-描述流程實現端到端(end-to-end)訓練的一個主要障礙是, 關鍵點偵測步驟通常包含一個不可微分(non-differentiable)的操作, 即尋找局部最大值(argmax)。這個操作會阻斷梯度在反向傳播過程中的流動, 使得偵測器無法透過梯度下降進行優化<sup>43</sup>。
  - 早期解決方案:
    - **TILDE**: 該方法繞過了直接優化偵測器的難題, 轉而訓練一個網路來回歸一個「偵測器響應熱力圖」, 使其在不同光照條件下(如白天和黑夜)能夠模仿傳統偵測器的穩定輸出<sup>40</sup>。
    - **LIFT**: 這是首個真正意義上的端到端學習流程, 它將偵測、方向賦值和描述子生成三個環節整合在一個統一的框架中進行聯合訓練<sup>31</sup>。
    - **LF-Net**: 在 LIFT 的基礎上進行了改進, 採用了更統一的架構和自監督學習策略, 從而減少了對帶有三維真值的數據集的依賴<sup>43</sup>。

### 3.3 前沿範式: 自監督與無偵測器模型

近年來, 兩種革命性的範式將學習特徵推向了新的高度。

#### SuperPoint: 自監督的偵測與描述

- 核心思想: SuperPoint 巧妙地解決了為真實世界圖像標註大量高質量關鍵點的難題。由於「興趣點」的定義本身存在模糊性, 人類標註者難以可靠、一致地完成這項工作<sup>44</sup>。SuperPoint 採用了一種自監督(self-supervised)的方法, 讓模型自己為自己生成訓練標籤<sup>31</sup>。
- 訓練流程:
  1. 合成數據預訓練: 首先, 在一個包含大量簡單幾何形狀(如立方體、線條、星星等)的合成數據集上訓練一個基礎偵測器, 稱為「MagicPoint」。在這個合成數據集中, 角點的真值位置是精確已知的<sup>44</sup>。
  2. 單應性自適應(Homographic Adaptation): 這是 SuperPoint 的關鍵創新。為了給真實圖像生成「偽真值」(pseudo-ground truth)標籤, 演算法對一張真實圖像應用大量的隨機單應性變換(模擬不同的相機視角), 生成許多扭曲後的圖像。然後, 在所有這些扭曲圖像上運行預訓練好的 MagicPoint 偵測器。最後, 將偵測到的關鍵點透過單應性變換的逆變換, 映射回原始圖像的坐標系。透過聚合來自多個不同視角的偵測結果, 就可以為原始真實圖像生成一套穩定且高質量的偽真值關鍵點標籤<sup>44</sup>。
  3. 聯合訓練: 最後, 使用這個帶有偽真值標籤的大規模真實圖像數據集, 來訓練一個最終的、統一的 SuperPoint 網路。該網路擁有一個共享的編碼器和兩個解碼器, 一個

用於關鍵點偵測，另一個用於描述子生成。透過這種方式，偵測器和描述子被協同優化，從而能夠在真實世界的圖像上表現出色<sup>44</sup>。

## LoFTR：基於 Transformer 的無偵測器匹配

- 「無偵測器」革命：LoFTR (Local Feature TRansformer) 徹底顛覆了傳統的「先偵測，後描述」的範式<sup>31</sup>。它的動機源於傳統偵測器在低紋理或無特徵區域（如白牆、天空）的根本性失敗——在這些區域，根本不存在穩定可重複的「興趣點」<sup>31</sup>。
- 架構與機制：
  1. 從粗到精 (**Coarse-to-Fine**)：LoFTR 首先在一個較粗糙的特徵層級（例如，圖像尺寸的 1/8）上建立像素級的密集匹配，然後在一個精細的層級上對這些匹配進行優化<sup>31</sup>。
  2. **Transformer** 提供全局上下文：LoFTR 的核心是 Transformer 模組。它不再處理局部圖像塊，而是將兩幅圖像的特徵圖展平，並輸入到一系列自注意力 (**self-attention**) 和交叉注意力 (**cross-attention**) 層中<sup>32</sup>。
  3. 注意力機制如何工作：自注意力層允許每個像素的特徵從其自身所在圖像的全局範圍內聚合上下文資訊；而交叉注意力層則允許每個像素的特徵去檢視另一幅圖像的全局範圍，以尋找其對應點。這種全局感受野 (global receptive field) 的能力，使得 LoFTR 能夠在無紋理區域成功匹配。例如，一個位於白牆上的點，可以利用房間角落、傢俱輪廓等遠處的上下文資訊來確定自己的身份並找到匹配<sup>32</sup>。
- 性能表現：LoFTR 在發布之時，其性能遠超當時所有的特徵匹配方法，特別是在具有挑戰性的室內和室外場景中，迅速成為了新的技術標竿<sup>31</sup>。其主要缺點是 Transformer 帶來的較高計算複雜度，因此後續的研究工作多集中於提升其運行效率<sup>47</sup>。

## 範式比較：手工設計特徵 vs. 學習特徵

下表從多個維度對手工設計特徵和基於學習的特徵這兩種範式進行了根本性的對比，以揭示該領域的演進脈絡。

維度	手工設計特徵 (Hand-crafted Features)	基於學習的特徵 (Learned Features)
代表演算法	SIFT, SURF, ORB	SuperPoint, LoFTR
設計哲學	基於人類對幾何和物理的先驗知識，設計被認為是魯棒的數學規則 <sup>41</sup> 。	定義一個可優化的匹配目標（損失函數），讓神經網路從數據中自動學習達成該目標的最優特徵表示 <sup>39</sup> 。
數據依賴性	無需訓練數據。演算法參數是固定的。	需要大規模、多樣化的訓練數據集。性能與數據質量和數量直接相關。
理想條件下性能	性能穩定可靠，尤其是在紋理豐富、結構清晰的場景中。	性能極佳，通常在各項指標上超越手工方法 <sup>45</sup> 。

挑戰性場景性能	在低紋理、極端光照變化和劇烈天氣影響下性能會顯著下降 <sup>31</sup> 。	魯棒性更強，特別是像 LoFTR 這樣的模型，能夠處理手工方法完全失效的場景 <sup>31</sup> 。
計算成本	推理速度快(特別是 ORB)，無需訓練。	訓練過程計算成本極高。推理速度因模型而異，LoFTR 等模型計算量較大 <sup>47</sup> 。
可解釋性	可解釋性強。我們能明確知道演算法為何選擇某個點作為特徵(例如，因為它是 DoG 極值點)。	可解釋性較差(「黑箱」問題)。很難直觀理解網路學習到的高維特徵表示的具體含義。
核心範式	先偵測，後描述。	SuperPoint 沿用「偵測-描述」流程但以學習方式實現；LoFTR 則開創了「無偵測器」的密集匹配範式。

這張表格揭示了該領域的演進並非簡單的「新舊取代」，而是一場深刻的哲學變革。手工特徵的設計理念是：「讓我們基於我們認為穩健的幾何原理來定義特徵」。而學習特徵的理念是：「讓我們定義一個匹配的目標，然後讓網路自己去探索什麼樣的特徵最有效」。LoFTR 的「無偵測器」架構更是對 SIFT 乃至 SuperPoint 所遵循的「先偵測」哲學的直接挑戰和顛覆。這表明，電腦視覺的發展並非一條直線，而是對問題不同層面、不同假設的持續探索與反思。

## 第四部分：應用、預處理與未來展望

幾何特徵的提取與匹配並非純粹的理論研究，它們是眾多電腦視覺應用得以實現的核心技術。本部分將透過具體的案例來展示其應用價值，並探討相關的必要預處理步驟以及未來的發展方向。

### 4.1 幾何特徵的實際應用

#### 案例研究一：圖像拼接(全景圖生成)

圖像拼接是將多張具有重疊區域的圖像合成為一張無縫、高解析度的全景圖的過程，是幾何特徵最經典的應用之一<sup>48</sup>。

一個典型的圖像拼接流程如下：

1. 圖像採集：拍攝一組視角連續且有足夠重疊區域的圖像<sup>48</sup>。
2. 特徵偵測與描述：對每對相鄰的圖像，使用如 ORB 這樣快速的演算法來提取關鍵點和描述子<sup>48</sup>。選擇 ORB 是因為其在速度和性能之間取得了良好的平衡，且無專利限制。
3. 特徵匹配：使用暴力匹配(Brute-Force)或 FLANN 等方法，基於描述子之間的相似度(對 ORB 而言是漢明距離)來尋找初始的匹配對。通常會結合「比例測試」(Ratio Test)來過濾掉一部分模糊的匹配<sup>26</sup>。

4. 使用 RANSAC 估計單應性矩陣：上一步得到的匹配對中必然包含外點。此時，將這些匹配對作為 RANSAC 演算法的輸入，以魯棒地估計出一個  $3 \times 3$  的單應性矩陣 ( $H$ )。這個矩陣描述了將一個圖像平面上的點映射到另一個圖像平面所需的透視變換<sup>50</sup>。這是整個流程中至關重要的外點剔除步驟。
5. 圖像變形 (Warping)：利用估算出的單應性矩陣  $H$ ，對其中一張圖像進行透視變換，使其與另一張圖像的視角對齊<sup>48</sup>。
6. 融合與渲染：將變形後的圖像與基準圖像進行融合。為了消除拼接處的明顯接縫，通常會採用羽化 (feathering) 或多頻段融合 (multi-band blending) 等技術，最終生成平滑自然的全景圖<sup>48</sup>。

在實際開發中，像 OpenCV 這樣的函式庫既提供了實現上述每一步的底層函數（如 `cv2.ORB_create`, `cv2.findHomography`），也提供了一個封裝了完整流程的高階 `Stitcher` 類，極大地簡化了開發過程<sup>49</sup>。

## 案例研究二：即時定位與地圖構建 (SLAM)

SLAM (Simultaneous Localization and Mapping) 是指移動設備（如機器人、無人機、AR 眼鏡）在未知環境中，即時構建環境地圖的同時，確定自身在該地圖中位置的技術<sup>54</sup>。這是實現自主導航的基礎<sup>56</sup>。

- 傳統幾何特徵的角色：經典的基於特徵的 SLAM 系統（如 ORB-SLAM）完全建立在點、線等幾何特徵之上<sup>54</sup>。它們的作用體現在：
  - 運動跟蹤：透過匹配連續相機幀之間的特徵點（如 ORB 特徵），可以估算出相機的運動（平移和旋轉）<sup>54</sup>。
  - 地圖構建：將匹配上的特徵點進行三角化，計算出它們在三維空間中的位置，這些三維點共同構成了一個稀疏的點雲地圖<sup>57</sup>。
  - 迴路閉合 (Loop Closure)：當設備回到一個曾經訪問過的地方時，透過將當前看到的特徵與地圖中已有的所有特徵進行匹配，系統可以識別出「迴路」。一旦檢測到迴路，就可以利用這個資訊來校正長時間累積的定位誤差<sup>54</sup>。
- 向高階特徵的轉變：
  - 問題：傳統幾何特徵在動態環境中（如場景中有行人、車輛移動）或在光照劇變等挑戰性條件下非常脆弱<sup>54</sup>。一個移動的物體會產生不符合「靜態世界」假設的特徵軌跡，從而嚴重干擾定位的準確性。
  - 解決方案：物體級特徵：當前 SLAM 領域的前沿趨勢是使用更高層次的物體級特徵<sup>54</sup>。系統不再僅僅依賴於無意義的點和線，而是利用基於深度學習的物體偵測器，來識別和跟蹤場景中完整的物體（如椅子、桌子、汽車）。
  - 優勢：物體級特徵提供了遠比點、線更豐富的資訊。它們是更穩定的觀測單元，攜帶著語義資訊（例如，「這是一把椅子」），並能提供更強的幾何約束（例如，長方體、對稱性等先驗知識）用於後端優化<sup>54</sup>。這使得 SLAM 系統在動態場景中更為魯棒，並且能夠構建出包含語義資訊的地圖。這樣的地圖不僅用於定位，還能更好地支持人機互動和高階任務規劃<sup>54</sup>。

## 4.2 不可或缺的預處理:相機校準

- 鏡頭畸變問題:真實的相機鏡頭並非理想的針孔模型,它們會引入幾何畸變,主要包括徑向畸變(導致圖像邊緣的直線看起來像桶形或枕形)和切向畸變<sup>58</sup>。
- 對特徵的影響:這些畸變會破壞圖像的幾何準確性。演算法偵測到的特徵點的(x,y)坐標,並非它們在理想針孔模型下的「真實」投影位置。對於需要高精度幾何測量的應用(如三維重建、視覺測量),直接使用這些帶有畸變的坐標會導致最終結果出現嚴重誤差。
- 解決方案:校準與去畸變:相機校準(Camera Calibration)就是估計相機內部參數(如焦距、主點)和畸變係數的過程。這通常是透過拍攝一張已知的校準板(如棋盤格)在不同姿態下的圖像來完成的<sup>58</sup>。一旦獲得了這些參數,就可以利用畸變係數對後續拍攝的每一張圖像進行「去畸變」(undistortion)處理,生成一張更接近理想針孔模型的圖像<sup>58</sup>。對於任何依賴高精度幾何資訊的視覺任務而言,相機校準都是一個至關重要的預處理步驟。

## 4.3 結論與未來展望

本報告引領讀者踏上了一段從像素到感知的旅程,見證了電腦視覺領域如何從基於幾何原理、由人類精心設計的 SIFT 時代,演進到由數據驅動、為任務優化的 SuperPoint 和 LoFTR 的學習時代。這段歷程的核心,始終圍繞著在精度、速度、魯棒性、計算成本和可解釋性之間的權衡。

展望未來,幾何特徵的發展將呈現以下幾個趨勢:

- 協同而非取代:未來並非是學習方法完全取代傳統方法。結合兩者優勢的混合式方法將愈發重要,例如,利用傳統偵測器的速度和可預測性,結合學習描述子的強大表示能力(如 SIFT 偵測器 + HardNet 描述子),可以在性能和效率之間取得新的平衡<sup>42</sup>。
- 跨越二維:特徵匹配的原理正在被擴展到跨過多模態數據的匹配任務中,例如將 RGB 圖像與三維雷射雷達點雲進行匹配,或對不同類型的聲納圖像進行配準。深度特徵融合與注意力機制在其中扮演著關鍵角色<sup>43</sup>。
- 無偵測器模型的效率提升:一個核心的研究方向是提升像 LoFTR 這樣的無偵測器模型的運行效率,使其能夠應用於更大規模、更要求即時性的場景,如大規模三維重建和圖像檢索<sup>47</sup>。
- 與高階語義的深度融合:正如在 SLAM 應用中看到的趨勢,未來的特徵匹配將不僅僅是尋找像素級的對應關係,而是建立場景中具有語義意義的部件之間的對應。未來的「特徵」可能不再是一個點或一個圖像塊,而是與整個場景、特定任務和先驗知識相關聯的動態實體<sup>54</sup>。
- 新興計算範式的整合:正如一些前瞻性研究所指出的,未來特徵提取技術可能會與量子計算、神經形態架構等新興技術相結合,以實現更高的效率;或與聯邦學習結合,以在保護數據隱私的前提下進行模型訓練<sup>41</sup>。

回顧整個領域的發展,可以發現一個有趣的循環:從抽象化到上下文的再整合。以 SIFT 為代表的傳統演算法,其核心思想是將圖像抽象為一組稀疏的局部關鍵點,為了效率和局部魯棒性,刻意捨棄了全局的上下文資訊。然而,這種抽象化在低紋理等場景中恰恰成為了其弱點,因為在這些場景中,被捨棄的全局上下文對於消除歧義至關重要<sup>31</sup>。而以 LoFTR 為代表的現代方法,則利用 Transformer 的強大能力,明確地將全局上下文

再整合回特徵表示中。其自注意力和交叉注意力機制，就是為了讓每個點的特徵能夠感知到兩幅圖像的完整內容<sup>32</sup>。這標誌著從純粹的局部抽象，向全局感知的回歸。這表明，該領域的演進並非一條單向的直線，而是一個不斷在局部精確性和全局理解之間尋求更優平衡的螺旋式上升過程。未來的突破，或許就蘊藏在如何更智慧地融合局部與全局資訊，實現真正意義上的、從像素到感知的跨越。

## 引用的著作

1. Mastering Feature Extraction in Computational Geometry - Number Analytics, 檢索日期:8月 2, 2025,  
<https://www.numberanalytics.com/blog/mastering-feature-extraction-computational-geometry>
2. Geometric Modeling for Computer Vision - DTIC, 檢索日期:8月 2, 2025,  
<https://apps.dtic.mil/sti/tr/pdf/ADA002261.pdf>
3. 计算机视觉中的传统特征提取方法总结 - 极市, 檢索日期:8月 2, 2025,  
<https://www.cvmart.net/community/detail/5869>
4. en.wikipedia.org, 檢索日期:8月 2, 2025,  
[https://en.wikipedia.org/wiki/Geometric\\_feature\\_learning#:~:text=Geometric%20features%20are%20features%20of,detected%20by%20feature%20detection%20methods.](https://en.wikipedia.org/wiki/Geometric_feature_learning#:~:text=Geometric%20features%20are%20features%20of,detected%20by%20feature%20detection%20methods.)
5. Geometric feature learning - Wikipedia, 檢索日期:8月 2, 2025,  
[https://en.wikipedia.org/wiki/Geometric\\_feature\\_learning](https://en.wikipedia.org/wiki/Geometric_feature_learning)
6. 详解SIFT,SURF,ORB,FAST 特征提取算法比较 - 华为云, 檢索日期:8月 2, 2025,  
<https://bbs.huaweicloud.com/blogs/420685>
7. DEF: Deep Estimation of Sharp Geometric Features in 3D Shapes - NYU Courant, 檢索日期:8月 2, 2025, <https://cims.nyu.edu/gcl/papers/2022-DEF.pdf>
8. Improved ORB Algorithm Using Three-Patch Method and Local Gray Difference - PMC, 檢索日期:8月 2, 2025, <https://pmc.ncbi.nlm.nih.gov/articles/PMC7070684/>
9. Local Feature Matching Using Deep Learning: A Survey - arXiv, 檢索日期:8月 2, 2025, <https://arxiv.org/html/2401.17592v2>
10. Illumination Scale Rotation | PDF | Vision | Artificial Intelligence - Scribd, 檢索日期:8月 2, 2025, <https://www.scribd.com/document/571203159/Chpt-6-SIFT>
11. www.geeksforgeeks.org, 檢索日期:8月 2, 2025,  
<https://www.geeksforgeeks.org/computer-vision/describe-the-concept-of-scale-invariant-feature-transform-sift/#:~:text=What%20is%20Scale%2DInvariant%20Feature,affine%20transformations%20and%20illumination%20changes.>
12. SIFT, SURF,ORB,FAST,BRIEF特征提取算法的比较原创 - CSDN博客, 檢索日期:8月 2, 2025, [https://blog.csdn.net/weixin\\_38570251/article/details/82628080](https://blog.csdn.net/weixin_38570251/article/details/82628080)
13. Comparison of Feature Detection and Matching Approaches: SIFT and SURF, 檢索日期:8月 2, 2025,  
[https://www.researchgate.net/publication/314285930\\_Comparison\\_of\\_Feature\\_Detection\\_and\\_Matching\\_Approaches\\_SIFT\\_and\\_SURF](https://www.researchgate.net/publication/314285930_Comparison_of_Feature_Detection_and_Matching_Approaches_SIFT_and_SURF)
14. Scale and Rotation invariant feature descriptors - Signal Processing Stack Exchange, 檢索日期:8月 2, 2025,  
<https://dsp.stackexchange.com/questions/4893/scale-and-rotation-invariant-feat>

## ure-descriptors

15. Words as Geometric Features: Estimating Homography using Optical Character Recognition as Compressed Image Representation - arXiv, 檢索日期:8月 2, 2025, <https://arxiv.org/html/2505.18925v1>
16. 详解SIFT、SURF和ORB特征点检测和描述算法\_stitch python surf sift ..., 檢索日期:8月 2, 2025, <https://blog.csdn.net/MRZHUGH/article/details/133669449>
17. ORB: an efficient alternative to SIFT or SURF, 檢索日期:8月 2, 2025, <https://mzucker.github.io/swarthmore/papers/rublee2011orb.pdf>
18. Full comparison results between ORB, SURF and SIFT - ResearchGate, 檢索日期:8月 2, 2025, [https://www.researchgate.net/figure/Full-comparison-results-between-ORB-SURF-and-SIFT\\_tbl2\\_354998356](https://www.researchgate.net/figure/Full-comparison-results-between-ORB-SURF-and-SIFT_tbl2_354998356)
19. The Ultimate ORB Guide - Number Analytics, 檢索日期:8月 2, 2025, <https://www.numberanalytics.com/blog/the-ultimate-orb-guide>
20. (PDF) ORB: an efficient alternative to SIFT or SURF - ResearchGate, 檢索日期:8月 2, 2025, [https://www.researchgate.net/publication/221111151\\_ORB\\_an\\_efficient\\_alternative\\_to\\_SIFT\\_or\\_SURF](https://www.researchgate.net/publication/221111151_ORB_an_efficient_alternative_to_SIFT_or_SURF)
21. ALGD-ORB: An improved image feature extraction algorithm with adaptive threshold and local gray difference | PLOS One - Research journals, 檢索日期:8月 2, 2025, <https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0293111>
22. Mastering ORB in Computer Vision - Number Analytics, 檢索日期:8月 2, 2025, <https://www.numberanalytics.com/blog/ultimate-guide-to-orb-in-computer-vision>
23. INVARIANCE - MICC, 檢索日期:8月 2, 2025, [http://www.micc.unifi.it/delbimbo/wp-content/uploads/2011/03/slide\\_corso/A31%20feature%20invariance.pdf](http://www.micc.unifi.it/delbimbo/wp-content/uploads/2011/03/slide_corso/A31%20feature%20invariance.pdf)
24. Rotation and Translation Invariant Representation Learning with Implicit Neural Representations - arXiv, 檢索日期:8月 2, 2025, <https://arxiv.org/pdf/2304.13995>
25. Towards Rotation Invariance in Object Detection - CVF Open Access, 檢索日期:8月 2, 2025, [https://openaccess.thecvf.com/content/ICCV2021/papers/Kalra\\_Towards\\_Rotation\\_Invariance\\_in\\_Object\\_Detection\\_ICCV\\_2021\\_paper.pdf](https://openaccess.thecvf.com/content/ICCV2021/papers/Kalra_Towards_Rotation_Invariance_in_Object_Detection_ICCV_2021_paper.pdf)
26. Mastering Feature Matching Techniques - Number Analytics, 檢索日期:8月 2, 2025, <https://www.numberanalytics.com/blog/mastering-feature-matching-techniques-algorithms-implementations>
27. Feature Matching Across 1D Panoramas - Computer Science, 檢索日期:8月 2, 2025, <https://www.cs.middlebury.edu/~schar/papers/omnivis05.pdf>
28. Object Detection Dilemmas: Conquering Occlusions, Scale, and Pose Variations | by Jing li, 檢索日期:8月 2, 2025, <https://medium.com/@jingli0155/object-detection-dilemmas-conquering-occlusions-scale-and-pose-variations-08d8e888b69e>
29. Results of line matching on a pair of images with occlusion. This... - ResearchGate, 檢索日期:8月 2, 2025,

[https://www.researchgate.net/figure/Results-of-line-matching-on-a-pair-of-images-with-occlusion-This-figure-is-better-viewed\\_fig2\\_221363700](https://www.researchgate.net/figure/Results-of-line-matching-on-a-pair-of-images-with-occlusion-This-figure-is-better-viewed_fig2_221363700)

30. SOLD2: Self-supervised Occlusion-aware Line Description and Detection - Microsoft, 檢索日期: 8月 2, 2025,  
<https://www.microsoft.com/en-us/research/publication/sold2-self-supervised-occlusion-aware-line-description-and-detection/>
31. LoFTR: Detector-Free Local Feature Matching With Transformers - CVF Open Access, 檢索日期: 8月 2, 2025,  
[https://openaccess.thecvf.com/content/CVPR2021/papers/Sun\\_LoFTR\\_Detector-Free\\_Local\\_Feature\\_Matching\\_With\\_Transformers\\_CVPR\\_2021\\_paper.pdf](https://openaccess.thecvf.com/content/CVPR2021/papers/Sun_LoFTR_Detector-Free_Local_Feature_Matching_With_Transformers_CVPR_2021_paper.pdf)
32. LoFTR: Detector-Free Local Feature Matching with Transformers, 檢索日期: 8月 2, 2025, <https://zju3dv.github.io/loftr/>
33. Random sample consensus - Wikipedia, 檢索日期: 8月 2, 2025,  
[https://en.wikipedia.org/wiki/Random\\_sample\\_consensus](https://en.wikipedia.org/wiki/Random_sample_consensus)
34. IMPROVING RANSAC FEATURE MATCHING BASED ON GEOMETRIC RELATION, 檢索日期: 8月 2, 2025, <https://d-nb.info/1239825420/34>
35. RANSAC - MATLAB & Simulink - MathWorks, 檢索日期: 8月 2, 2025,  
<https://www.mathworks.com/discovery/ransac.html>
36. The Ultimate Guide to the RANSAC Algorithm - Think Autonomous, 檢索日期: 8月 2, 2025, <https://www.thinkautonomous.ai/blog/ransac-algorithm/>
37. Outlier detection using the RANSAC algorithm | by saurabh dasgupta - Medium, 檢索日期: 8月 2, 2025,  
<https://medium.com/@saurabh.dasgupta1/outlier-detection-using-the-ransac-algorithm-de52670adb4a>
38. RANSAC for Computer Vision - Number Analytics, 檢索日期: 8月 2, 2025,  
<https://www.numberanalytics.com/blog/ransac-for-computer-vision>
39. A deep learning-based local feature extraction method for improved image matching and surface reconstruction from Yutu-2 PCAM images on the Moon | Request PDF - ResearchGate, 檢索日期: 8月 2, 2025,  
[https://www.researchgate.net/publication/375697057\\_A\\_deep\\_learning-based\\_local\\_feature\\_extraction\\_method\\_for\\_improved\\_image\\_matching\\_and\\_surface\\_reconstruction\\_from\\_Yutu-2\\_PCAM\\_images\\_on\\_the\\_Moon](https://www.researchgate.net/publication/375697057_A_deep_learning-based_local_feature_extraction_method_for_improved_image_matching_and_surface_reconstruction_from_Yutu-2_PCAM_images_on_the_Moon)
40. A survey of deep-learning-based image matching algorithms - DOI, 檢索日期: 8月 2, 2025, <https://doi.org/10.1117/12.3067778>
41. A Comprehensive Survey of Deep Learning Approaches in Image Processing - MDPI, 檢索日期: 8月 2, 2025, <https://www.mdpi.com/1424-8220/25/2/531>
42. A Coarse-to-Fine Feature Match Network Using Transformers for Remote Sensing Image Registration - MDPI, 檢索日期: 8月 2, 2025,  
<https://www.mdpi.com/2072-4292/15/13/3243>
43. Modality-Aware Feature Matching: A Comprehensive Review of Single- and Cross-Modality Techniques - arXiv, 檢索日期: 8月 2, 2025,  
<https://arxiv.org/html/2507.22791v1>
44. SuperPoint: Self-Supervised Interest Point Detection and ..., 檢索日期: 8月 2, 2025  
,  
<https://sarawathimamidala30.medium.com/superpoint-self-supervised-interest->

## [point-detection-and-description-7d6b7b0ccf57](#)

45. Deep learning-based feature point detection (using SuperPoint). - ResearchGate, 檢索日期:8月 2, 2025,  
[https://www.researchgate.net/figure/Deep-learning-based-feature-point-detection-using-SuperPoint\\_fig9\\_378316290](https://www.researchgate.net/figure/Deep-learning-based-feature-point-detection-using-SuperPoint_fig9_378316290)
46. Coarse TRVO: A Robust Visual Odometry with Detector-Free Local Feature - Fuji Technology Press, 檢索日期:8月 2, 2025,  
<https://www.fujipress.jp/jaciii/jc/jacii002600050731/>
47. Efficient LoFTR: Semi-Dense Local Feature Matching with Sparse-Like Speed - arXiv, 檢索日期:8月 2, 2025, <https://arxiv.org/html/2403.04765v1>
48. Image Stitching with OpenCV - GeeksforGeeks, 檢索日期:8月 2, 2025,  
<https://www.geeksforgeeks.org/computer-vision/image-stitching-with-opencv/>
49. Image Stitching in OpenCV, 檢索日期:8月 2, 2025,  
<https://www.opencvhelp.org/tutorials/advanced/image-stitching/>
50. Image Stitching with OpenCV and Python - PyImageSearch, 檢索日期:8月 2, 2025  
,  
<https://pyimagesearch.com/2018/12/17/image-stitching-with-opencv-and-python/>
51. How to stitch two images using homography matrix in OpenCv? - Stack Overflow, 檢索日期:8月 2, 2025,  
<https://stackoverflow.com/questions/61146241/how-to-stitch-two-images-using-homography-matrix-in-opencv>
52. AgRowStitch: A High-fidelity Image Stitching Pipeline for Ground-based Agricultural Images, 檢索日期:8月 2, 2025, <https://arxiv.org/html/2503.21990v1>
53. High level stitching API (Stitcher class) - OpenCV Documentation, 檢索日期:8月 2, 2025, [https://docs.opencv.org/4.x/d8/d19/tutorial\\_stitcher.html](https://docs.opencv.org/4.x/d8/d19/tutorial_stitcher.html)
54. Higher or Lower: Challenges in Object based SLAM, 檢索日期:8月 2, 2025,  
<https://arxiv.org/pdf/2310.13256>
55. YPR-SLAM: A SLAM System Combining Object Detection and Geometric Constraints for Dynamic Scenes - PMC, 檢索日期:8月 2, 2025,  
<https://pmc.ncbi.nlm.nih.gov/articles/PMC11511238/>
56. Image matching application scenarios. SLAM, simultaneous localization and mapping - ResearchGate, 檢索日期:8月 2, 2025,  
[https://www.researchgate.net/figure/Image-matching-application-scenarios-SLA-M-simultaneous-localization-and-mapping\\_fig2\\_378316290](https://www.researchgate.net/figure/Image-matching-application-scenarios-SLA-M-simultaneous-localization-and-mapping_fig2_378316290)
57. SLAM-OR: Simultaneous Localization, Mapping and Object Recognition Using Video Sensors Data in Open Environments from the Sparse Points Cloud - MDPI, 檢索日期:8月 2, 2025, <https://www.mdpi.com/1424-8220/21/14/4734>
58. Camera Calibration, Camera Undistortion, Fisheye Camera, & Pinhole Camera | Python (OpenCV) & MATLAB (The Computer Vision Toolbox™) | by Nassim | Medium, 檢索日期:8月 2, 2025,  
<https://medium.com/@nassim.hammami98/camera-calibration-camera-undistortion-fisheye-camera-pinhole-camera-python-opencv-7d697cfb591e>
59. Local Features and Image Matching - Kornia - Read the Docs, 檢索日期:8月 2, 2025, <https://kornia.readthedocs.io/en/latest/feature.html>

60. Deep Learning-Based Feature Matching Algorithm for Multi-Beam and Side-Scan Images, 檢索日期:8月2, 2025, <https://www.mdpi.com/2072-4292/17/4/675>