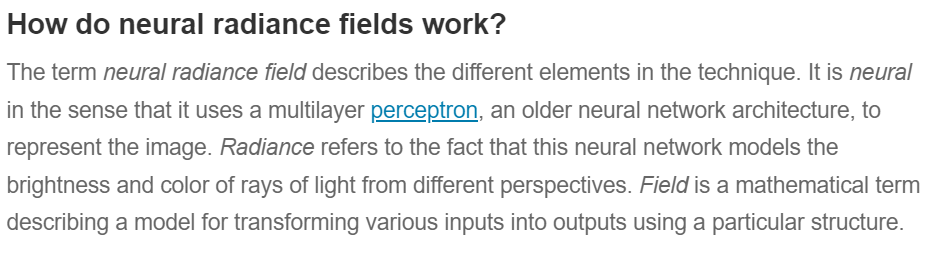
# NeRF ([Neural radiance field](https://en.wikipedia.org/wiki/Neural_radiance_field))

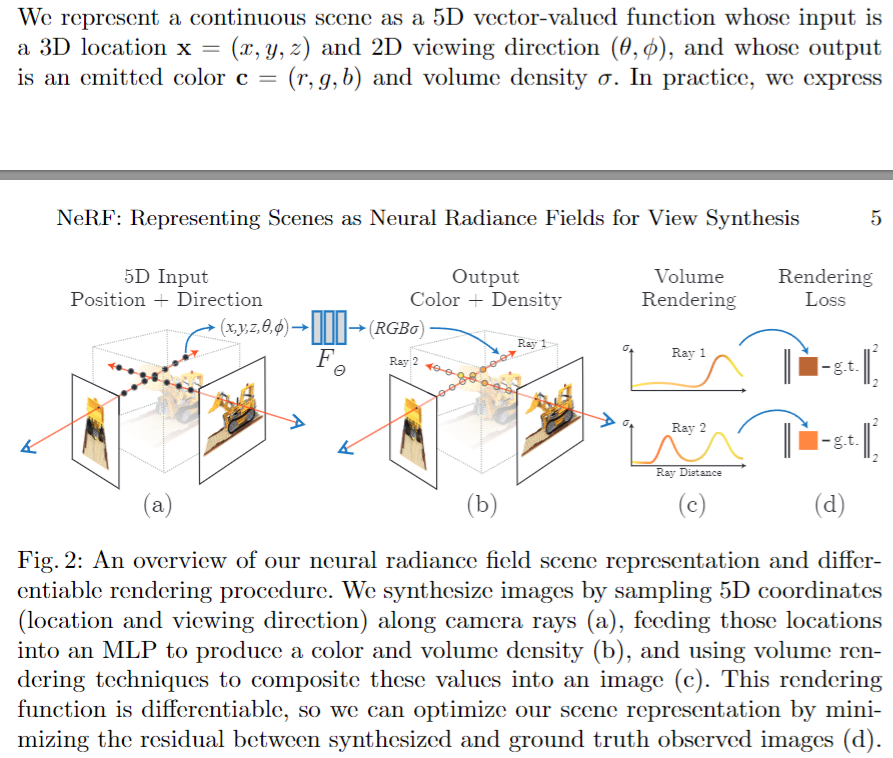
[](https://www.techtarget.com/searchenterpriseai/definition/neural-radiance-fields-NeRF)

NeRF khác các kỹ thuật DL khác ở chỗ các ảnh dùng để train 1 mạng fully connected NN chỉ dùng để generate các views khác cho 1 cảnh nhất định.

Mô hình NeRF (Neural Radiance Field) hoạt động dựa trên nguyên tắc sử dụng mạng nơ-ron kết nối đầy đủ để tái tạo một cảnh 3D từ các ảnh 2D. Mạng này sẽ dự đoán **màu sắc và mật độ thể tích** cho mỗi điểm trong không gian 3D, dựa trên vị trí điểm và hướng nhìn của camera.

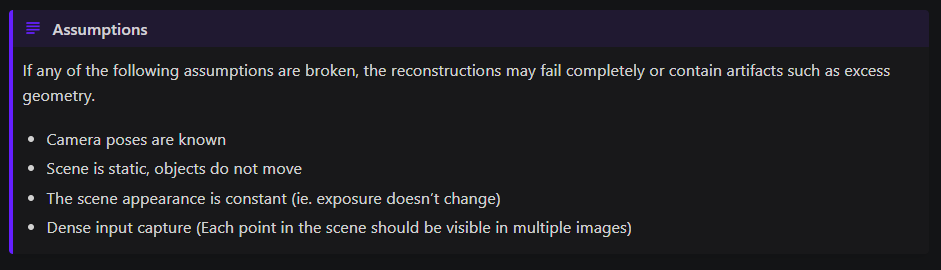
# NeRF hoạt động như thế nào?

[**NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis**](https://arxiv.org/pdf/2003.08934)



## 1. Input

* **Tập hợp ảnh 2D của cảnh:** Đây là dữ liệu đầu vào chính của NeRF, bao gồm nhiều ảnh chụp từ các góc nhìn khác nhau.
* **Vị trí và hướng camera (camera pose) tương ứng với mỗi ảnh:** Thông tin này rất quan trọng để mạng nơ-ron hiểu được mối quan hệ không gian giữa các ảnh và cảnh 3D. (chạy bằng SfM, thường là COLMAP)
  + **Sfm:** Doppelgangers là 1 phần quan trọng trong bài toán này.

Việc dựng 3D sẽ không thực hiện được nếu có lỗi trong khâu dữ liệu structure from motion. (Assumptions cap từ màn hình Method của Nerfstudio).

* **Giới hạn cảnh:** Xác định không gian 3D mà NeRF sẽ biểu diễn. Đối với dữ liệu tổng hợp, giới hạn này thường là một khối lập phương chứa toàn bộ cảnh. Đối với dữ liệu thực, tọa độ chuẩn hóa thiết bị được sử dụng để ánh xạ phạm vi độ sâu của các điểm vào [-1, 1].

## 2. Output

* **Mật độ (σ):** Thể hiện mức độ chắn sáng của mỗi điểm trong không gian 3D. Mật độ cao cho biết điểm đó mờ đục, mật độ thấp cho biết điểm đó trong suốt.
* **Màu sắc (RGB):** Thể hiện màu sắc của mỗi điểm trong không gian 3D khi được quan sát từ một hướng nhìn cụ thể.
* **Pháp tuyến bề mặt (tuỳ chọn):** Một số mô hình NeRF cũng có thể dự đoán pháp tuyến bề mặt, giúp cải thiện chất lượng hình học của kết xuất.

## 3. Cấu trúc mô hình

Mô hình NeRF thường được cấu thành từ các thành phần sau:

* **Mã hóa vị trí (Positional Encoding):** Ánh xạ tọa độ 3D (x, y, z) và hướng nhìn (θ, φ) thành vector đặc trưng có chiều cao hơn. Việc mã hóa này sử dụng các hàm tần số cao để giúp mạng nơ-ron học được các biểu diễn chi tiết hơn.
* **Mạng nơ-ron (MLP - Multi-Layer Perceptron):** Nhận vector đặc trưng từ mã hóa vị trí làm đầu vào và xuất ra mật độ và màu sắc. MLP thường bao gồm nhiều lớp kết nối đầy đủ với hàm kích hoạt ReLU.
* **Kết xuất thể tích (Volume Rendering):** Tính toán màu sắc cuối cùng của pixel dựa trên mật độ và màu sắc dọc theo tia nhìn. Quá trình này tích lũy màu sắc và mật độ từ các điểm lấy mẫu dọc theo tia nhìn, tương tự như kỹ thuật alpha compositing.
* **Lấy mẫu phân cấp (Hierarchical Sampling):** Kỹ thuật lấy mẫu hiệu quả giúp giảm thiểu số lượng điểm cần được đánh giá bởi mạng nơ-ron. Mô hình NeRF thường sử dụng kết hợp bộ lấy mẫu đồng nhất và bộ lấy mẫu PDF, tập trung vào các vùng có khả năng chứa bề mặt của vật thể.

# Cấu trúc NeRF Studio

Nerfstudio là một framework mã nguồn mở được xây dựng dựa trên PyTorch, cung cấp một cách tiếp cận **mô-đun hóa** để phát triển các phương pháp NeRF (Neural Radiance Fields), cho phép **hiển thị thời gian thực** và dễ dàng sử dụng với dữ liệu thực tế. Cấu trúc của Nerfstudio có thể được chia thành các thành phần chính sau:

## DataManager:

* + Chịu trách nhiệm chuyển đổi các hình ảnh có vị trí thành **RayBundles**.
  + **DataParser:**
    - Nằm trong DataManager, có nhiệm vụ tải hình ảnh đầu vào và dữ liệu camera.
    - Tương thích với nhiều định dạng dữ liệu như COLMAP, Record3D, Polycam, KIRI Engine, Metashape, và Reality Capture.
  + Sau khi hình ảnh được tải và định dạng, DataManager lặp qua dữ liệu, tạo RayBundles và dữ liệu giám sát (ground truth).
  + Cũng có thể tối ưu hóa vị trí camera trong quá trình huấn luyện.

## RayBundles, RaySamples và Frustums:

* + **RayBundles:** Biểu diễn một lát cắt không gian 3D, được tham số hóa bởi điểm gốc, hướng, và thông tin meta như chỉ số camera và thời gian.
  + **RaySamples:** Được tạo ra từ RayBundles bằng cách xác định khoảng cách giữa các bins, đại diện cho các phần không gian 3D được lấy mẫu dọc theo mỗi tia.
  + **Frustums:** Đại diện cho các phần không gian 3D được lấy mẫu, có thể được mã hóa dưới dạng điểm mẫu hoặc Gaussian với giá trị trung bình và hiệp phương sai.

## Models và Fields:

* + RayBundles được đưa vào Models, sau đó được lấy mẫu thành RaySamples.
  + RaySamples được Fields sử dụng để chuyển đổi các vùng không gian 3D (Frustums) thành các đại lượng như màu sắc hoặc mật độ.
  + Nerfstudio cung cấp nhiều **lựa chọn triển khai** cho Models và Fields, bao gồm:
    - **Mã hóa đặc trưng:** Fourier features, hash encodings, spherical harmonics, và matrix decompositions.
    - **Thành phần Field:** Fused MLPs, voxel grids, surface normal MLPs, hàm kích hoạt, spatial distortions, và temporal distortions.

## Trình hiển thị Web thời gian thực:

* + Được xây dựng dựa trên ReactJS, cho phép hiển thị NeRF trong thời gian thực trong quá trình huấn luyện hoặc thử nghiệm.
  + Truyền phát **dòng video WebRTC** từ phiên huấn luyện NeRF đến trình duyệt web, cho phép người dùng di chuyển camera và xem kết quả trực tiếp.
  + Cung cấp nhiều **tính năng**, bao gồm:
    - Chuyển đổi giữa các đầu ra mô hình (RGB, độ sâu, pháp tuyến, ngữ nghĩa).
    - Tạo đường dẫn camera tùy chỉnh.
    - Hiển thị hình ảnh huấn luyện trong không gian 3D.
    - Cắt và xuất ra point cloud và mesh.

## Xuất hình học:

* + Hỗ trợ xuất ra các định dạng như point cloud, TSDF (truncated signed distance function) để tạo mesh, và Poisson surface reconstruction.

## Phương pháp Nerfacto:

* + Là phương pháp **mặc định** của Nerfstudio, kết hợp các ý tưởng từ nhiều bài báo nghiên cứu để đạt được sự cân bằng giữa tốc độ và chất lượng.
  + Sử dụng **kỹ thuật lấy mẫu phân cấp** với 3 bước:
    - Tối ưu hóa góc nhìn camera.
    - Lấy mẫu đồng nhất đến một khoảng cách cố định từ camera.
    - Sử dụng mạng proposal để lấy mẫu tập trung vào các vùng quan trọng.
  + Áp dụng **co rút cảnh** để nén không gian vô hạn thành một hộp giới hạn có kích thước cố định.
  + Sử dụng **mã hóa băm** (hash encoding) với **lưới băm** (hash grid) để biểu diễn hàm mật độ của cảnh.
  + Tích hợp **nhúng ngoại hình cho mỗi hình ảnh** để xử lý sự khác biệt về độ phơi sáng giữa các camera.
  + Tính toán và dự đoán **pháp tuyến bề mặt**.

# Mối quan hệ giữa Doppelgangers Matching và NeRF

**Doppelgangers Matching** là bài toán phân biệt các cặp hình ảnh tương tự nhau về mặt thị giác xem chúng có miêu tả cùng một bề mặt 3D hay không. **NeRF (Neural Radiance Fields)** là một kỹ thuật sử dụng mạng nơ-ron để biểu diễn trường rạng rỡ của một cảnh 3D, từ đó tạo ra ảnh từ góc nhìn mới.

Mặc dù hai phương pháp này giải quyết hai bài toán khác nhau, chúng có **mối liên hệ** đáng chú ý:

* **Vấn đề Doppelgangers là thách thức đối với NeRF:** Doppelgangers (hình ảnh "bản sao") có thể gây khó khăn cho việc huấn luyện và kết xuất NeRF.
  + Nếu tập dữ liệu huấn luyện NeRF chứa các cặp hình ảnh Doppelgangers được gánh nhãn sai (như là cùng một bề mặt), mô hình NeRF có thể học sai lệch, dẫn đến kết quả tái tạo 3D không chính xác.
  + Ví dụ, NeRF có thể tạo ra các cấu trúc hình học "ma" (ghost structures) do nhầm lẫn các bề mặt tương tự nhau trong Doppelgangers.
* **Doppelgangers Matching có thể cải thiện hiệu suất của NeRF:**
  + Bằng cách xác định và loại bỏ các cặp Doppelgangers khỏi tập dữ liệu huấn luyện, ta có thể giúp NeRF học được biểu diễn chính xác hơn về cảnh 3D.
  + Doppelgangers Matching cũng có thể được sử dụng như một bước xử lý trước (pre-processing) để lọc bỏ các cạnh sai trong biểu đồ cảnh (scene graph) trước khi áp dụng NeRF, từ đó cải thiện chất lượng tái tạo 3D.
* **Cả hai phương pháp đều dựa trên feature matching:**
  + Doppelgangers Matching là bài toán feature matching trên các cặp hình ảnh tương tự nhau.
  + NeRF, mặc dù không trực tiếp sử dụng feature matching để tái tạo 3D, nhưng việc **mã hóa vị trí** trong NeRF (phần **SfM**) có thể được xem như một dạng feature matching ở mức độ trừu tượng hơn. Cả hai kỹ thuật này đều nhằm mục đích biểu diễn thông tin về cảnh 3D thông qua các đặc trưng thị giác.

Doppelgangers Matching giúp **giải quyết** một thách thức quan trọng đối với NeRF, trong khi NeRF cung cấp một **công cụ** hiệu quả để tái tạo 3D sau khi các Doppelgangers đã được xử lý.

# ----> ADDING STRUCTURE FROM MOTION