Borrador resumen de Depth Prediction Evaluation

Capítulo 1 al 4:

"UniDepthV2: Estimación de Profundidad Métrica Monocular Universal Simplificada"

Se presenta un nuevo modelo llamado **UniDepthV2**, el cual está diseñado para la estimación de profundidad métrica monocular (MMDE). Este modelo tiene la capacidad de reconstruir escenas **3D** métricas que se generan a partir de imágenes individuales, abordando las limitaciones de los métodos **MMDE** existentes, los cuales a menudo fallan en generalizar a dominios no vistos durante el entrenamiento.

UniDepthV2 se centra en la tarea de **MMDE**, la cual es crucial para la comprensión de la estructura geométrica en diversas aplicaciones como la modelización en **3D**. Este modelo incluye un módulo de cámara auto promocionable que predice una señal no paramétrica para condicionar las características del módulo de profundidad. Esta representación se separa de la profundidad mediante una salida **pseudoesférica**, facilitando la optimización. Para asegurar la robustez de las características de profundidad ante variaciones de la cámara, se incorpora una pérdida de invarianza geométrica. Adicionalmente, una pérdida guiada por bordes mejora la precisión, localización y nitidez de los límites en la profundidad métrica estimada.

La capacidad de **UniDepthV2** de producir una salida de nivel de incertidumbre es esencial para determinar la confiabilidad de las predicciones del modelo en tareas futuras. Este modelo representa un avance significativo en la estimación de profundidad métrica monocular (**MMDE**) al ofrecer una solución que no solo estima la profundidad métrica de forma directa, sino que también exhibe una generalización superior a escenarios no vistos, gracias a sus innovadoras representaciones de cámara y estructura de salida, junto a sus funciones de pérdida optimizadas.

Fórmula:

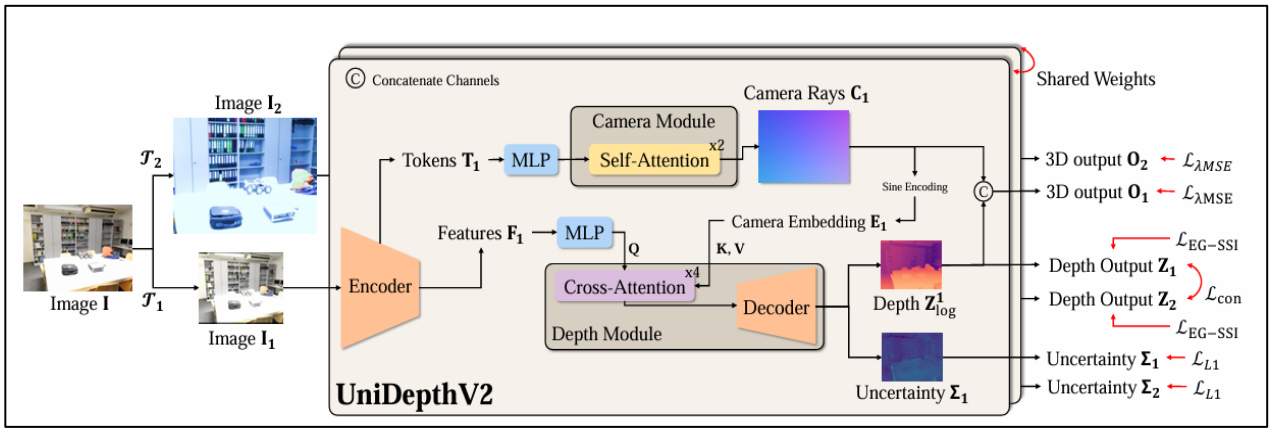
La pérdida de $\mathcal{L}_{EG-SSI}$ se define como:

$$\mathcal{L}_{EG-SSI}(\boldsymbol{D}, \boldsymbol{D}*, \Omega) = \sum_{\omega \in \Omega} \| \boldsymbol{N}_{\omega}(\boldsymbol{D}_{\omega}) - \boldsymbol{N}_{\omega}(\boldsymbol{D}^{*}_{\omega}) \|_{\mathbf{1}}$$

Otra fórmula relevante, aunque secundaria, es la **Geometric Invariance Loss** ($\mathcal{L}_{con}$)**:**

$$\mathcal{L}_{con}(Z_1, Z_2) = ||\mathcal{T}_2 \circ \mathcal{T}_1^{-1} \circ (Z_1) - sg(Z_2)||_1$$

Esquema:



Capítulo 5 al 8:

"Estimación progresiva de la profundidad monocular en el dominio del coseno discreto"

Este artículo presenta un nuevo enfoque para la estimación de profundidad monocular (**MDE**) llamado **DCDepth**. En lugar de la estimación de profundidad pixel a pixel en el dominio espacial, **DCDepth** estima los coeficientes de frecuencia de los parches de profundidad después de transformarlos al dominio del coseno discreto (**DCT**). Esta formulación permite modelar las correlaciones de profundidad local dentro de cada parche.

Este enfoque estima coeficientes de frecuencia de parches de profundidad, capturando correlaciones locales y descomponiendo la información en componentes de baja frecuencia (**estructura global**) y alta frecuencia (**detalles locales**).

Por una parte, la estrategia progresiva de **DCDepth** comienza prediciendo coeficientes de baja frecuencia para establecer un contexto global, refinando iterativamente los detalles locales mediante coeficientes de mayor frecuencia. Por su parte, la **DCT** inversa reconstruye el mapa de profundidad espacial. Su arquitectura incluye un codificador de imágenes, un módulo de fusión de características piramidales (**PFF**) con submuestreo basado en **DCT** para integrar características multiescalar, un decodificador para recuperar resolución y una cabeza de predicción progresiva (**PPH**) que usa **GRU** y atención cruzada para estimaciones iterativas.

El DCT descompone la información de profundidad en componentes de frecuencia, donde los componentes de baja frecuencia capturan la estructura central de la escena y los componentes de alta frecuencia detallan los aspectos más finos. DCDepth utiliza una estrategia progresiva, comenzando con la predicción de componentes de baja frecuencia para establecer un contexto de escena global, y luego refina los detalles locales mediante la predicción de componentes de frecuencia más alta.

Fórmula:

Es "La transformación de los coeficientes de frecuencia" que son predichos en el dominio de la transformada discreta del coseno (DCT) de vuelta al dominio espacial para obtener el mapa de profundidad estimado.

$$\widehat{\mathcal{D}} = T^{-1}(\psi(\mathcal{L}))$$

Otra fórmula relevante, es la función de pérdida total utilizada para entrenar el modelo:

$$L = L_d + \alpha * L_f + \beta * L_s$$

De donde podemos explorar cada parámetro de la siguiente manera:

L_d: Pérdida invariante a la escala (scale-invariant loss):

$$L_d = \alpha * \sum_{i=1}^{N} \beta^{N-i} \sqrt{\frac{1}{M}\sum d_i^2 - \frac{\lambda}{M^2}\left(\sum d_i\right)^2}$$

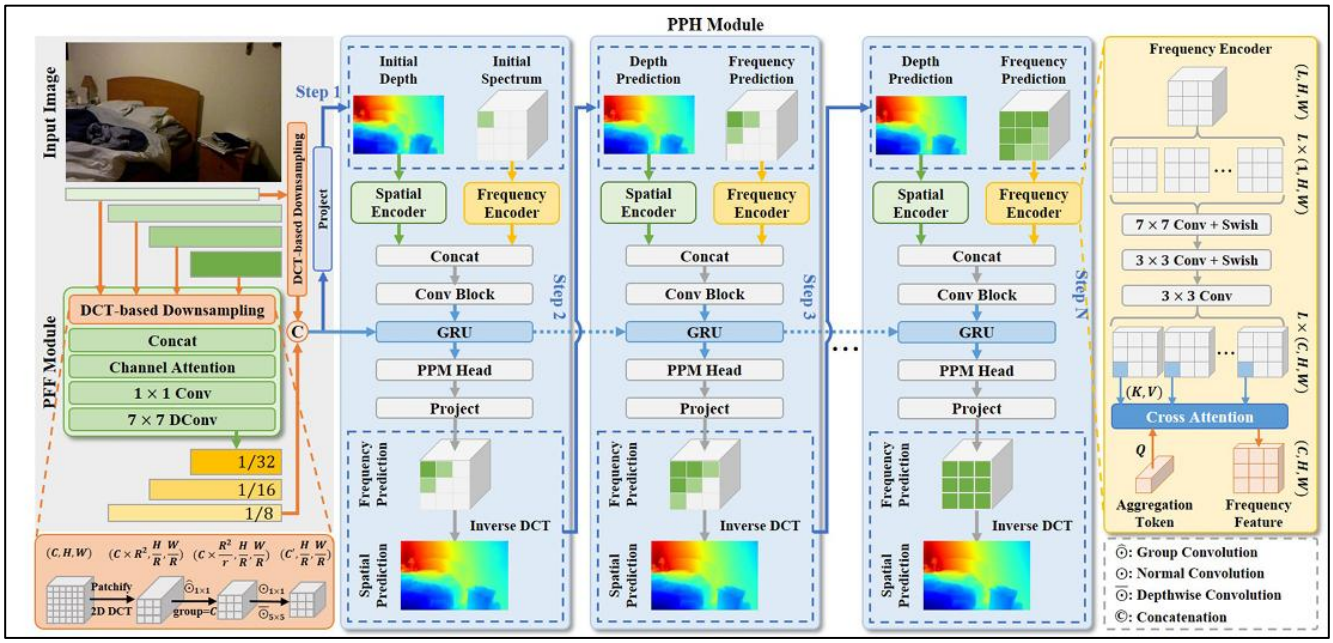
L_f: Pérdida de regularización de frecuencia para fomentar la dispersión de coeficientes de alta frecuencia

$$L_f = \sum (\epsilon^{u+v} - 1) * |f_{u,v}|$$

L_s: Pérdida de suavidad para promover mapas de profundidad suaves:

$$L_s = |\partial_x \widehat{\mathcal{D}}| * e^{-|\partial_x I_t|} + \left||\partial_y \widehat{\mathcal{D}}\right| * e^{-|\partial_y I_t|}$$

Esquema:



Capítulo 9

“Monocular asistido para distancia normal. Estimación de profundidad y finalización”

NDDepth, es un marco de aprendizaje profundo basado en geometría para la estimación y finalización de profundidad monocular (MDE), es fundamental para aplicaciones como robótica y conducción autónoma. Estima mapas intermedios de normales de superficie y distancias plano-origen, asumiendo que las escenas 3D están compuestas por planos fragmentados. Esta representación intermedia aprovecha la geometría de las escenas, mejorando la precisión al modelar regiones planas.

El marco integra dos cabezales, uno de distancia-normal para capturar estructuras planas y otro de profundidad regular para robustez en regiones de alta curvatura. Introduce una restricción de consistencia consciente del plano, aplicada a regiones detectadas mediante el algoritmo de segmentación de Felzenszwalb, fomentando uniformidad en normales y distancias dentro de planos. Para la estimación, un módulo de refinamiento iterativo contrastivo combina mapas de profundidad e incertidumbre. En la finalización, los mapas de ambos cabezales se fusionan usando incertidumbre y se refinan con una red de propagación espacial no local.

Su enfoque basado en física y segmentación planar lo distingue de métodos puramente basados en datos. NDDepth ofrece una base sólida para explorar MDE con enfoques geométricos, destacando por su precisión y aplicabilidad en escenas estructuradas, donde busca predecir el mapa de profundidad a partir de una sola imagen RGB.

Fórmula:

La más principal y fundamental es la que describe la relación entre la profundidad (D(p)), la normal de la superficie (N(p)) y la distancia del plano al origen (D(p)) para un punto **2D** p en la imagen.

$$D(p) = \frac{\mathcal{D}(p)}{N(p)K^{-1}\tilde{p}}$$

Su derivación proviene de:

Restricción normal-distancia:

$$\mathrm{N(p)P} = \mathcal{D}(\mathrm{p})$$

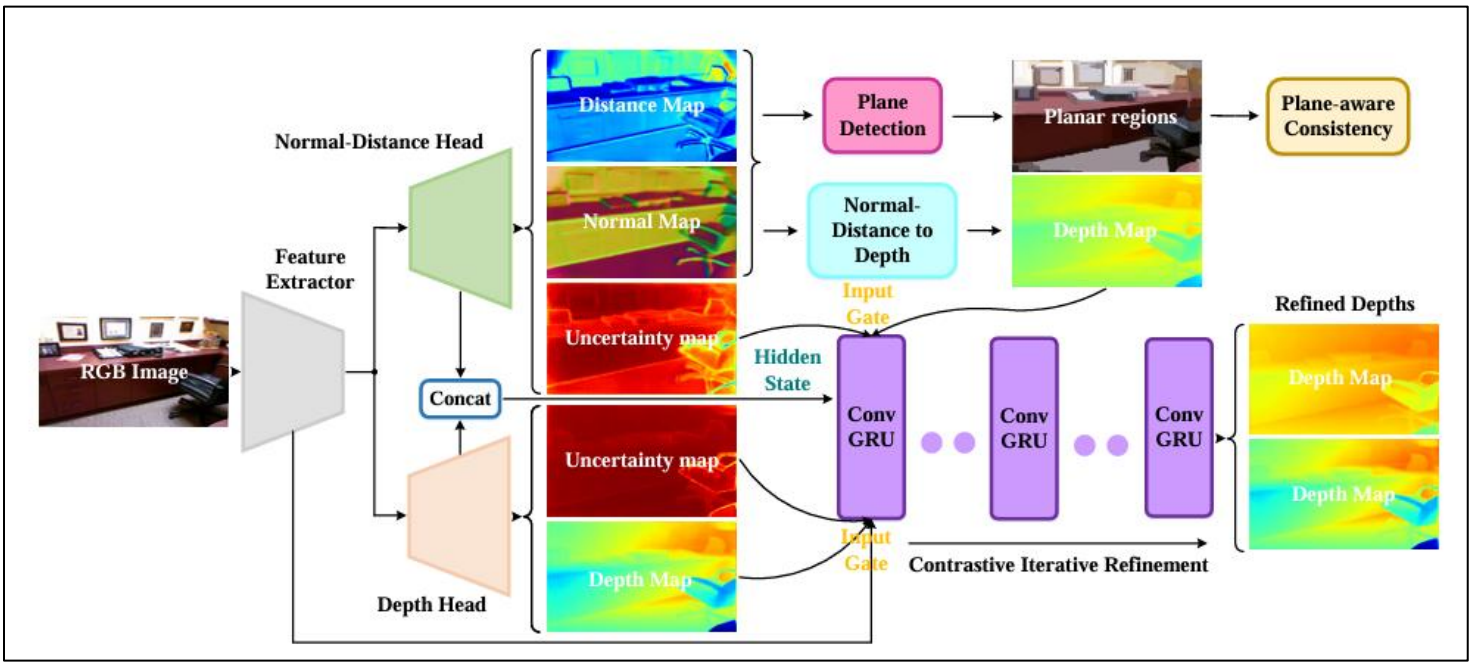
Proyección de la cámara pinole:

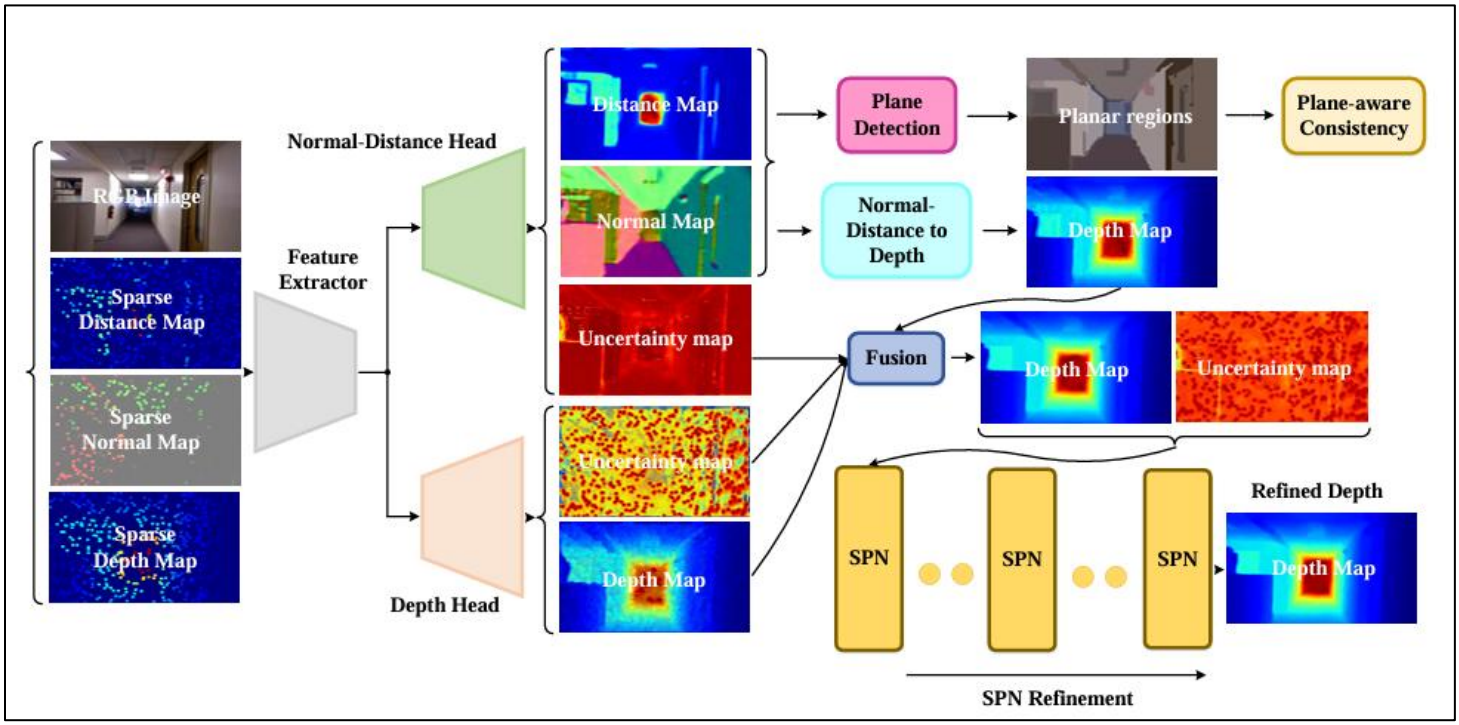
$$D(p)\tilde{p} = KP$$

Otra de las fórmulas más fundamentales dentro de este artículo es la Estimación de profundidad monocular:

$$\mathcal{L}_{pc} = \sum_{p} \mathcal{M}(p)|\nabla N(p)| + \sum_{p} \mathcal{M}(p)|\nabla \mathcal{D}(p)|$$

Esquema:


Normal-Distance Head
Feature Extractor
RGB Image
Distance Map
Normal Map
Uncertainty map
Plane Detection
Planar regions
Plane-aware Consistency
Normal-Distance to Depth
Depth Map
Input Gate
Refined Depths
Hidden State
Concat
Conv GRU
Conv GRU
Conv GRU
Depth Map
Depth Map
Uncertainty map
Depth Map
Input Gate
Contrastive Iterative Refinement
Depth Head


Normal-Distance Head
Feature Extractor
RGB Image
Sparse Distance Map
Sparse Normal Map
Sparse Depth Map
Distance Map
Normal Map
Uncertainty map
Plane Detection
Planar regions
Plane-aware Consistency
Normal-Distance to Depth
Depth Map
Fusion
Depth Map
Uncertainty map
Uncertainty map
Depth Map
SPN
SPN
SPN
Refined Depth
Depth Map
SPN Refinement
Depth Head

Capítulo 10

“IEBins: contenedores elásticos iterativos para Estimación de profundidad monocular”

El artículo presenta un enfoque novedoso para la estimación de profundidad monocular (MDE) basada en la clasificación-regresión. Los autores introducen Iterative Elastic Bins (IEBins), la cual es una estrategia que optimiza progresivamente el rango de búsqueda de profundidad a través de múltiples etapas. En cada una de las diferentes etapas se elabora una búsqueda más fina en el bin objetivo de la etapa anterior, ajustando el ancho de este bin de forma elástica según la incertidumbre de profundidad, calculada mediante la varianza de la distribución probabilística. Esto, a su vez, mitiga la acumulación de errores durante las iteraciones, mejorando la robustez y precisión.

El marco propuesto consta de un extractor de características, basado en Swin-Transformer, con un decodificador que usa módulos CRF neuronal para capturar correlaciones de largo alcance, y un optimizador iterativo basado en GRU, que modela el contexto temporal para predecir distribuciones probabilísticas por píxel. La predicción final combina linealmente los centros de los bins con estas distribuciones.

Se realizan contribuciones en los conjuntos de datos KITTI, NYU-Depth-v2 y SUN RGB-D, superando a competidores previos en métricas clave como Abs Rel y RMSE. Entre sus limitaciones se tiene que la suavidad de las predicciones puede difuminar bordes, y la falta de supervisión directa en la distribución probabilística podría afectar la precisión.

Fórmula:

La primordial para **IEBins** es la que describe la predicción de la profundidad $(\widehat{D}(p))$ a partir de una combinación lineal de candidatos de profundidad D_ny su distribución probabilística $(P_n(p))$para un pixel p.

$$\widehat{D}(p) = \sum_{n=0}^{N_{-1}} D_n * P_n(p)$$

De igual forma existe formulas de soporte entre las que destacan:

Cuantificación de la incertidumbre:

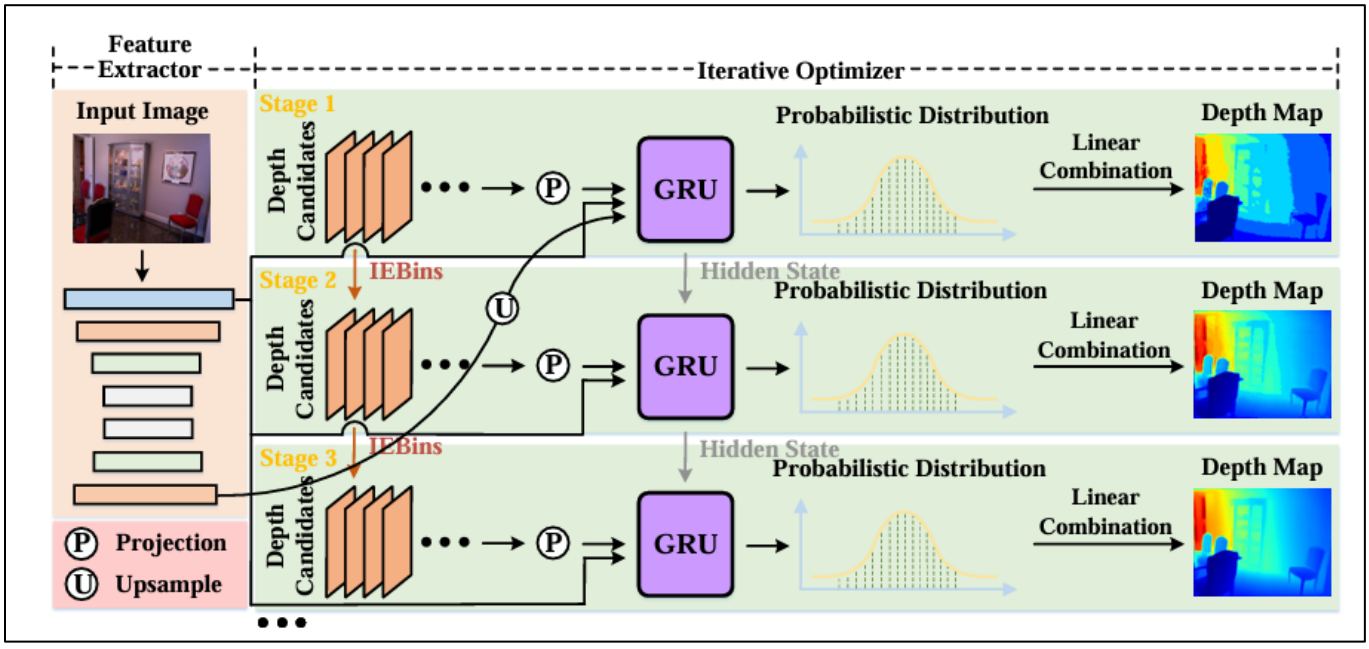
$$\widehat{\mathrm{V}}(\mathrm{p}) = \sum_{\mathrm{n}=0}^{\mathrm{N}-1} \left(\mathcal{D}_{\mathrm{n}} - \widehat{\mathrm{D}}(\mathrm{p})\right)^2 * \mathrm{P}_{\mathrm{n}}\,(\mathrm{p})$$

$$\widehat{\sigma}(\mathrm{p}) = \sqrt{\widehat{\mathrm{V}}(\mathrm{p})}$$

Función de pérdida de entrenamiento:

$$\mathcal{L}_{pixel} = \sum_{k=1}^{K} \alpha \sqrt{\frac{1}{|T|}\sum_{p}\big(g(p)\big)^2 - \frac{\beta}{|T|^2}\left(\sum_{p} g\,(p)\right)^2}$$

Esquema:


Feature Extractor
Iterative Optimizer
Input Image
Stage 1
Depth Candidates
GRU
Probabilistic Distribution
Linear Combination
Depth Map
Stage 2
IEBins
Hidden State
Stage 3
Projection
Upsample

Capítulo 11 – 12

“VA-DepthNet: un enfoque variacional para la predicción de la profundidad de una sola imagen”

Es un método innovador para la estimación de profundidad monocular (SIDP) que combina restricciones variacionales de primer orden con una red neuronal profunda para mejorar precisión y generalización. En lugar de predecir profundidades métricas píxel a píxel, modela gradientes de profundidad (Γx, Γy) y pesos de confianza (Σx, Σy), resolviendo un sistema matricial (ΣPZu = ΣΓ) para obtener un mapa de profundidad no escalado. Esto fomenta la regularidad espacial y preserva detalles de alta frecuencia.

La arquitectura de red basada en encoder-decoder utiliza un codificador Swin-Large para extraer características, un V-layer que predice 16 canales de gradientes y pesos, generando mapas de profundidad a 1/16 de resolución, y módulos de refinamiento que operan en resoluciones crecientes (1/8, 1/4) mediante fusión de características. Un módulo métrico estima escala y desplazamiento global con capas totalmente conectadas. La red se entrena de forma supervisada con una pérdida combinada: una pérdida de profundidad invariante a escala y una pérdida variacional que compara gradientes predichos con el ground truth (λ = 0.1).

Evaluado en KITTI, NYU Depth V2 y SUN RGB-D, VA-DepthNet supera a métodos como AdaBins y NeWCRFs en métricas clave. Muestra robusta generalización en pruebas zero-shot y maneja datos LiDAR dispersos en KITTI. Ablaciones confirman que el V-layer mejora la precisión frente a capas convolucionales o de autoatención, especialmente en backbones débiles. Estas limitaciones incluyen la necesidad de supervisión completa, pérdida potencial de detalles finos en el V-layer inicial (mitigada por refinamiento) y alta complejidad computacional, optimizada al operar a baja resolución.

Fórmula:

La primordial para **VA-DepthNet** es la que describe la construcción de la restricción variacional para recuperar el mapa de profundidad no escalado predicción de la profundidad ($\widetilde{Z_u}$) a partir de las diferencias de primer orden y sus pesos de confianza.

$$\tilde{E}P\widetilde{Z_u} = \tilde{E}\tilde{\Gamma}$$

Con su solución óptima la cual minimiza el residuo, se deriva como

$$\tilde{Z}_u^* = \overbrace{(P^T\widetilde{E^2}P)^{-1}}^{K_{\tilde{E}}} P^T\widetilde{E^2}\tilde{\Gamma}$$

Esquema:

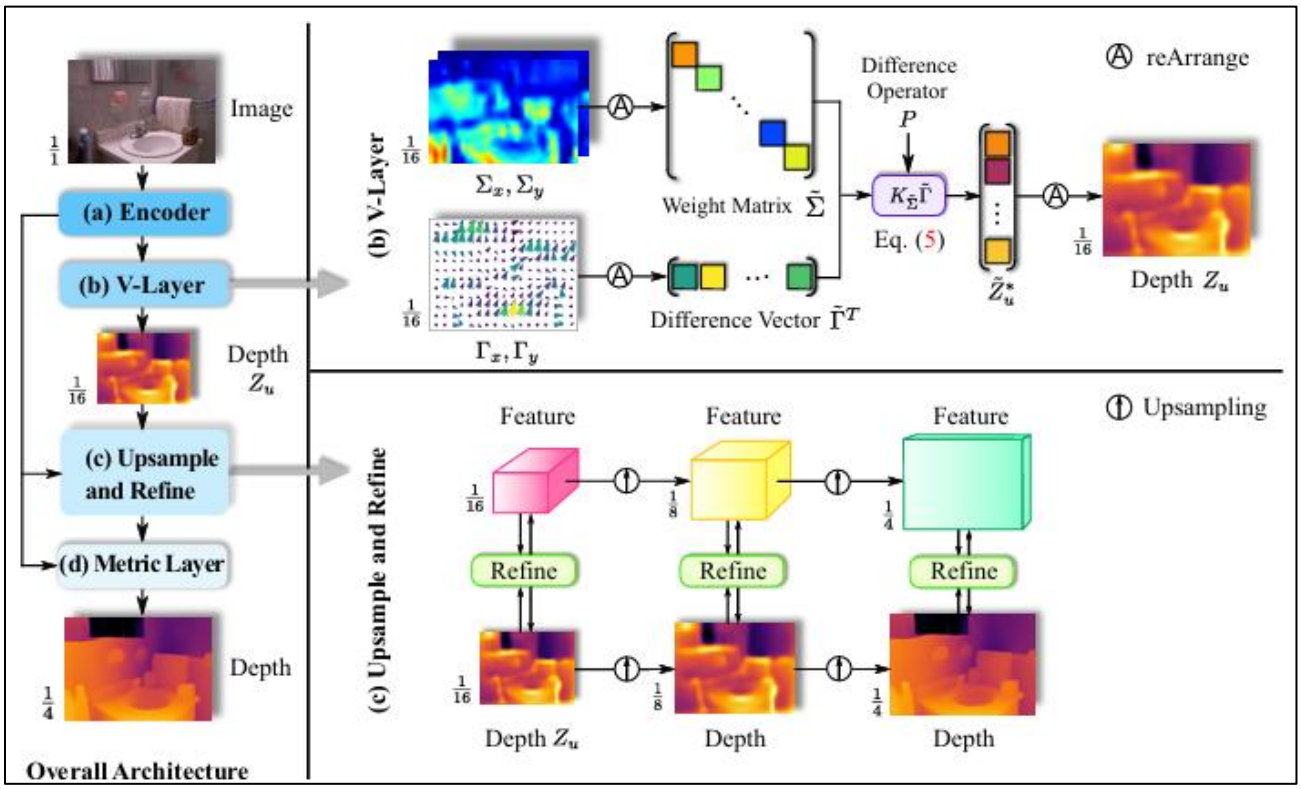


Image
(a) Encoder
(b) V-Layer
Depth Z_u
(c) Upsample and Refine
(d) Metric Layer
Depth
Overall Architecture
(b) V-Layer
Σ_x, Σ_y
Weight Matrix $\tilde{\Sigma}$
Difference Operator P
Eq. (5)
reArrange
Depth Z_u
Γ_x, Γ_y
Difference Vector $\tilde{\Gamma}^T$
(c) Upsample and Refine
Feature
Refine
Upsampling
Depth Z_u
Depth
Depth

Capítulo 13

"DiffusionDepth: Enfoque de eliminación de ruido por difusión para la estimación de la profundidad monocular"

DiffusionDepth reformula la estimación de profundidad monocular (SIDP) como un proceso de difusión-denoising, refinando iterativamente una distribución de profundidad desde ruido aleatorio, guiada por condiciones visuales monoculares. Opera en un espacio latente codificado por un codificador-decodificador de profundidad, donde el decodificador incluye una convolución 1x1, una deconvolución 3x3, una convolución 3x3 y una activación sigmoidea, transformando el latente refinado (x_0) en un mapa de profundidad. Usa un backbone Swin Transformer para extraer características multiescala, agregadas mediante una red piramidal (FPN) y HAHI para mejorar correlaciones globales y locales. El bloque de denoising condicionado (MCDB) fusiona condiciones visuales con latentes de profundidad mediante suma elemento a elemento, capas CNN y atención, optimizado con pérdidas combinadas (L_ddim, L_pixel, L_latent).

Para abordar la escasez de ground truth (GT) en KITTI (3.75-5% densidad), propone self-diffusion, agregando ruido a profundidades refinadas en lugar de GT dispersos, evitando el colapso modal con aumentos como recorte aleatorio y jitter. Estos a su vez logran resultados SOTA en KITTI y NYU-Depth-V2, superando a VA-Depth y URCDC-Depth, especialmente en interiores con GT denso, donde la difusión directa sobre GT también es viable. Las visualizaciones muestran bordes nítidos y formas claras frente a BinsFormer, mejorando detalles como postes de señalización.

Las limitaciones tienen métodos no difusivos, aunque optimizables reduciendo pasos (con leve pérdida de precisión). Ablaciones confirman que 20 pasos son óptimos, el self-diffusion es crucial para datos dispersos y el espacio latente de mayor resolución mejora resultados. Compatible con backbones como ResNet y ViT.

Fórmula:

La más principal y fundamental para DiffusionDepth es la que define el proceso de denoising condicionado para la estimación de profundidad monoculada, expresada como:

$$p_\theta(x_{t-1}|x_t) \coloneqq N(x_{t-1}; \mu_\theta(x_t, t), \sigma_t^2 I)$$

De igual manera, existen perdidas asociadas las cuales se representan mediante las siguientes formulas:

Pérdida de difusión (L_ddim):

$$\mathrm{L_{ddim}} = ||\mathrm{x_{t-1}} - \mu_\theta(\mathrm{x_t}, \mathrm{t}, \mathrm{c})||^2$$

Pérdida pixel-wise (L_pixel):

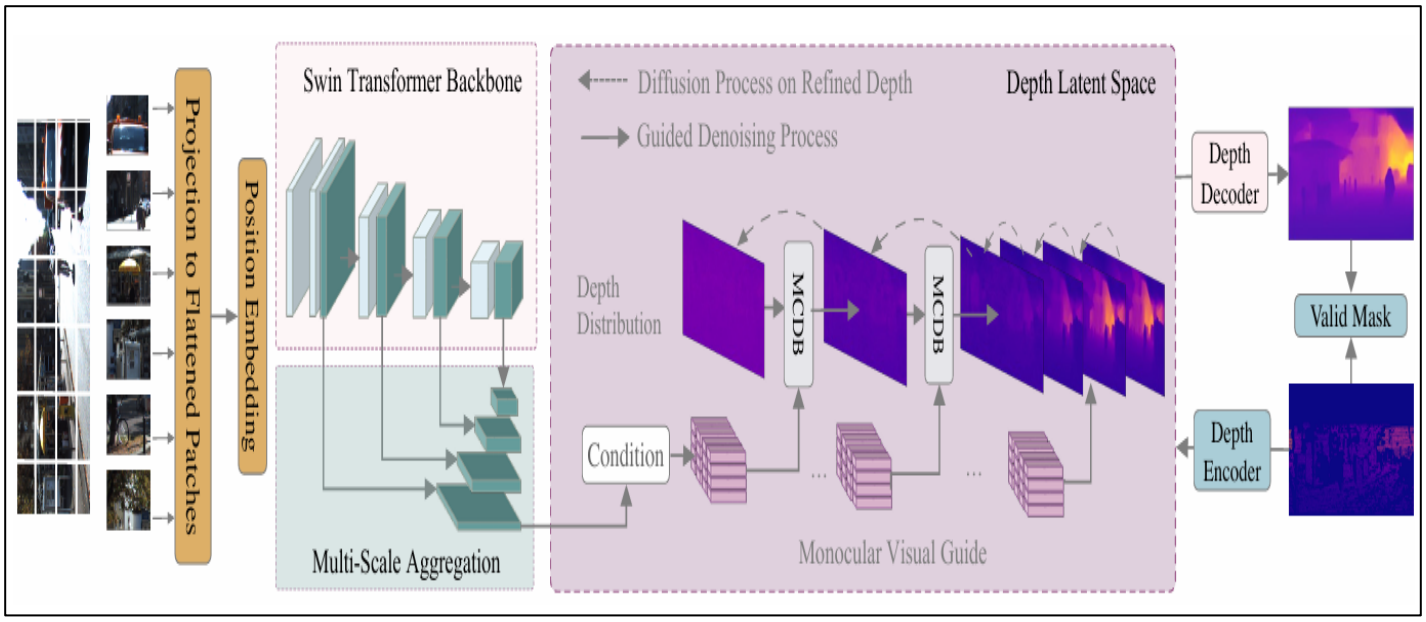
$$\mathrm{L_{pixel}} = \sqrt{\frac{1}{\mathrm{T}}\sum_{\mathrm{i}} \delta_{\mathrm{i}}^2 + \frac{\lambda}{\mathrm{T}^2}\left(\sum_{\mathrm{i}} \delta_{\mathrm{i}}\right)^2}$$

Pérdida en el espacio latente (L_latent) y pérdida total (L):

$$L_{latent} = \left|\left|x_0 - \hat{x}_0\right|\right|^2$$

$$L = \lambda_1 L_{ddim} + \lambda_2 L_{pixel} + \lambda_3 L_{pixel}$$

Esquema:


Projection to Flattened Patches
Position Embedding
Swin Transformer Backbone
Multi-Scale Aggregation
Diffusion Process on Refined Depth
Guided Denoising Process
Depth Latent Space
Depth Distribution
MCDB
MCDB
Condition
Monocular Visual Guide
Depth Decoder
Valid Mask
Depth Encoder

Capítulo 14

“iDisc: Internal Discretization for Monocular Depth Estimation”

El artículo presenta un nuevo enfoque para la estimación de profundidad monocular (MDE), un problema fundamental en visión basado por computadora. Los autores proponen el modelo iDisc, que introduce un módulo de Discretización Interna (ID) basado en un cuello de botella continuo-discreto-continuo. Este módulo a su vez descompone la escena en patrones de alto nivel (como objetos, planos y relaciones espaciales) sin imponer restricciones explícitas sobre la salida de profundidad, a diferencia de métodos previos que usan priors geométricos o discretización explícita del rango de profundidad.

El módulo ID utiliza atención cruzada "transpuesta" para una partición adaptativa de características (AFP), creando representaciones discretas internas (IDRs) que agrupan el espacio de características de manera dependiente de la entrada. Luego, la etapa de discretización interna de la escena (ISD) transfiere estas IDRs al espacio continuo mediante atención cruzada, generando mapas de profundidad.

iDisc logra resultados state-of-the-art en los conjuntos de datos NYU-Depth v2 y KITTI, superando a todos los métodos publicados en KITTI, y también en estimación de normales de superficie. Además, muestra una generalización robusta en pruebas de zero-shot en conjuntos como SUN-RGBD y Diode. Se abordan aspectos relevantes como la falta de diversidad en los conjuntos de datos al aire libre, proponiendo nuevos splits de Argoverse y DDAD para benchmarks más desafiantes.

Fórmula:

Se encuentra en la descripción del módulo de Internal Discretization (ID):

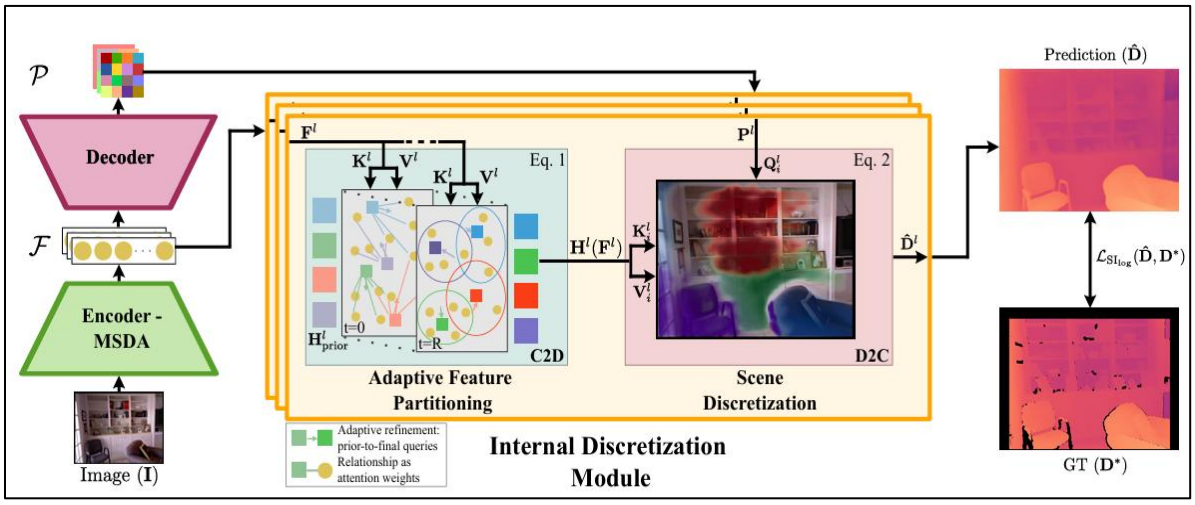
$$W_{ij}^{t} = \frac{\exp\left(k_i^T q_j^t\right)}{\sum_{k=1}^{N} exp\left(k_i^T q_k^t\right)}, q_j^{t+1} = \sum_{i=1}^{M} W_{ij}^{t} v_i$$

De igual manera se tiene la fórmula de la etapa discreto-a-continuo. Esta etapa transfiere las IDRs al espacio continuo de salida utilizando atención cruzada estándar.

La fórmula es:

$$D_{i+1} = softmax\left(Q_i K_i^T\right) V_i + D_i$$

Esquema:



Capítulo 15

“Single Image Depth Prediction Made Better: A Multivariate Gaussian Take”

El artículo propone un enfoque novedoso para la estimación de profundidad a partir de una sola imagen (SIDP) modelando la profundidad por píxel con una distribución gaussiana multivariada. A diferencia de métodos previos que predicen un valor escalar por píxel y asumen independencia entre píxeles, este modelo introduce una matriz de covarianza que captura la dependencia de la profundidad entre todos los píxeles de la imagen, reflejando relaciones espaciales en la escena.

Dado que calcular la covarianza completa es computacionalmente costoso, los autores proponen una aproximación de bajo rango para la matriz de covarianza, parametrizada por una red neuronal, reduciendo la complejidad de $O(N^3)$ a $O(NM+M^3)$, donde $M<<N$. La red se entrena usando una función de pérdida basada en la log-verosimilitud negativa (NLL), que generaliza pérdidas comunes como L2, escala-invariante y gradiente, unificando sus beneficios.

El modelo, denominado MG, logra resultados state-of-the-art en los conjuntos de datos NYU Depth V2, KITTI y SUN RGB-D, superando a métodos previos en métricas como SILog, Abs Rel y δ_1. Esto a su vez, indica una generalización robusta en pruebas zero-shot y proporciona una estimación de incertidumbre bayesiana mejorada al modelar dependencias entre píxeles. La implementación utiliza un codificador inspirado en Swin Transformer y decodificadores para estimar la media y covarianza, entrenados con el optimizador Adam.

Fórmula:

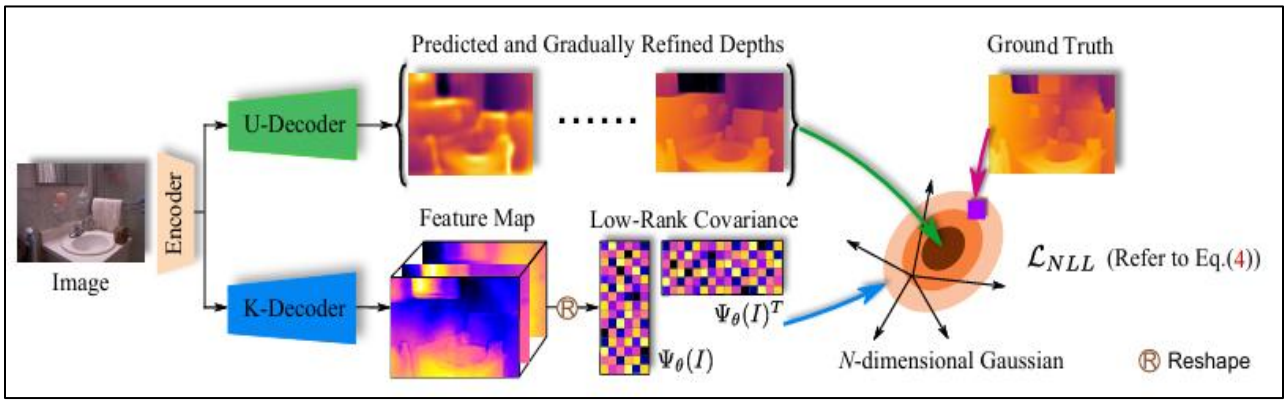
Es la que define el modelado de la distribución de profundidad como una distribución gaussiana multivariada con una matriz de covarianza parametrizada de bajo rango.

$$\log \Phi(Z|\theta, I) = -\frac{N}{2}\log 2\pi\sigma^2 - \frac{1}{2}\log\det(A) - \frac{\sigma^{-2}}{2}r^T r + \frac{\sigma^{-4}}{2}r^T \psi_\theta(I) A^{-1} \psi_\theta(I)^T r$$

Así mismo existe la ecuación de la función de pérdida total $\mathcal{L}_{NLL}$ para múltiples escalas y un término opcional de L2:

$$\mathcal{L}_{total} = \sum_{j=1}^{4} \mathcal{L}_{NLL}^{i} + \frac{1}{N}\sum_{i}(\mu_\theta^1(I)_i - Z_i^{gt})^2$$

Esquema:



Capítulo 16

"URCDC-Depth: Uncertainty Rectified Cross-Distillation with CutFlip for Monocular Depth Estimation"

El artículo presenta un marco novedoso para la estimación de profundidad monocular (MDE) que combina las fortalezas de un Transformer y una red neuronal convolucional (CNN) mediante destilación cruzada rectificada por incertidumbre. El modelo URCDC-Depth utiliza el Transformer para capturar correlaciones de largo alcance y la CNN para detalles locales, empleando las predicciones de profundidad de cada rama como pseudoetiquetas para entrenar a la otra. Para mitigar el impacto de etiquetas ruidosas, se modela la incertidumbre por píxel, ajustando los pesos de la pérdida en regiones inciertas. Además, se transfieren mapas de características del Transformer a la CNN, con unidades de acoplamiento que fusionan estas características, reduciendo la brecha de capacidad entre ambas ramas.

El modelo introduce una técnica de aumento de datos llamada **CutFlip**, que corta y voltea verticalmente las imágenes de entrenamiento para disminuir la dependencia del modelo en la posición vertical de la imagen, fomentando el uso de otras pistas de profundidad. La pérdida total combina una pérdida de escala invariante, la pérdida de destilación cruzada rectificada y la pérdida de incertidumbre.

URCDC-Depth supera a métodos previos en los conjuntos de datos KITTI, NYU-Depth-v2 y SUN RGB-D, mejorando métricas como Abs Rel, RMSE y SILog, sin carga computacional adicional en inferencia, ya que solo usa la rama Transformer. Dichos experimentos demuestran que la destilación cruzada, la incertidumbre, las unidades de acoplamiento y CutFlip mejoran significativamente la precisión. Este enfoque unifica la modelación global y local, ofreciendo una solución robusta y eficiente para MDE.

Fórmula:

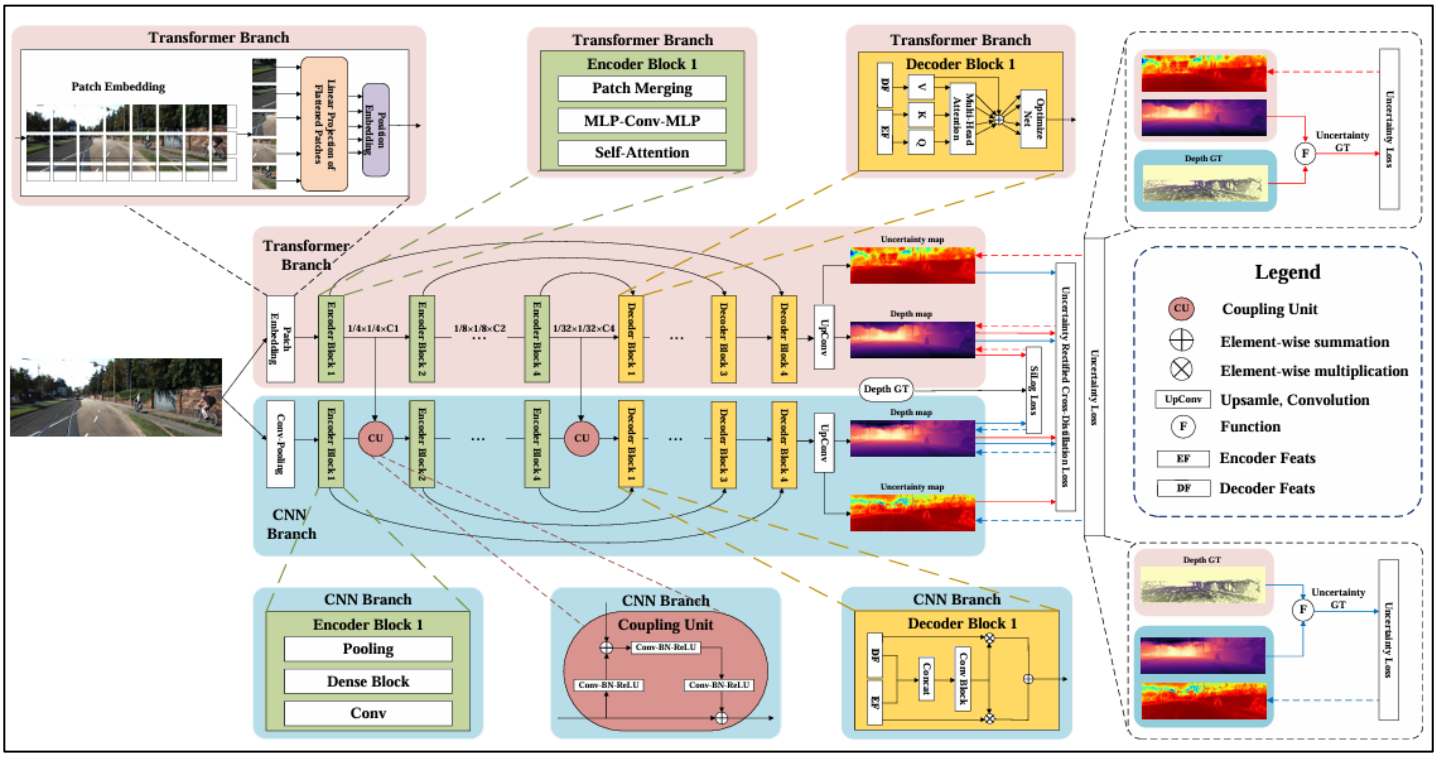
La que trata la pérdida de destilación cruzada rectificada por incertidumbre $\mathcal{L}_{urcd}$. Esta ecuación encapsula el núcleo de la contribución principal: un mecanismo de destilación cruzada entre una rama Transformer y una rama CNN

$$\mathcal{L}_{urcd} = \sum_{p} \left(1 - \overline{u_n^c}(p)\right) \odot | \, d_n^t(p) - \overline{d_n^c}(p) \, | + \sum_{p} \left(1 - \overline{u_n^t}(p)\right) \odot | \, d_n^c(p) - \overline{d_n^t}(p) |$$

También se tiene la pérdida invariante a escala (scaled scale-invariant loss), que mide la diferencia logarítmica entre la profundidad predicha y la profundidad real, definida como:

$$\mathcal{L}_{ssi} = \mathcal{K} \sqrt{\frac{1}{|T|} \sum_{p} (g_n^t(p))^2 - \frac{n}{|T|^2} (\sum_{p} g_n^t(p))^2} + \mathcal{K} \sqrt{\frac{1}{|T|} \sum_{p} (g_n^c(p))^2 - \frac{n}{|T|^2} (\sum_{p} g_n^c(p))^2} \, ; p \in T$$

Esquema:


Transformer Branch
Patch Embedding
Linear Projection of Flattened Patches
Position Embedding
Transformer Branch
Encoder Block 1
Patch Merging
MLP-Conv-MLP
Self-Attention
Transformer Branch
Decoder Block 1
DF
EF
V
K
Q
Multi-Head Attention
Optimize Net
Uncertainty GT
Depth GT
Uncertainty Loss
Transformer Branch
Patch Embedding
Encoder Block 1
1/4×1/4×C1
Encoder Block 2
1/8×1/8×C2
Encoder Block 4
1/32×1/32×C4
Decoder Block 1
Decoder Block 3
Decoder Block 4
UpConv
Uncertainty map
Depth map
Depth GT
SiLog Loss
Uncertainty Rectified Cross Distillation Loss
Uncertainty Loss
Conv-Pooling
CU
CNN Branch
Legend
Coupling Unit
Element-wise summation
Element-wise multiplication
UpConv
Upsamle, Convolution
F
Function
EF
Encoder Feats
DF
Decoder Feats
CNN Branch
Encoder Block 1
Pooling
Dense Block
Conv
CNN Branch
Coupling Unit
Conv-BN-ReLU
Conv-BN-ReLU
Conv-BN-ReLU
CNN Branch
Decoder Block 1
DF
EF
Concat
Conv Block
Depth GT
Uncertainty GT
Uncertainty Loss

Capítulo 17

“BinsFormer: Revisiting Adaptive Bins for Monocular Depth Estimation”

Este articulo propone un novedoso marco para la estimación de profundidad monocular basado en **clasificación-regresión,** destacando por su enfoque en la generación adaptativa de bins y la interacción entre distribuciones de probabilidad y predicciones de bins. El modelo, denominado BinsFormer, utiliza un decodificador Transformer para generar bins, tratándolo como un problema de predicción de conjunto a conjunto, y una estructura de decodificador multi-escala para estimar mapas de profundidad de manera gruesa a fina, capturando información geométrica espacial. Además, incorpora una consulta auxiliar de comprensión de escena que mejora la precisión al aprender información implícita mediante una tarea de clasificación ambiental.

Este framework supera los métodos previos al evitar problemas como la pérdida de información global y la interferencia entre representaciones de bins y píxeles. Emplea un módulo de píxeles para extraer características locales, un módulo Transformer para generar bins globales y un módulo de estimación de profundidad que combina ambos mediante una combinación lineal de centros de bins y distribuciones de probabilidad. La estrategia multi-escala y la tarea auxiliar de clasificación optimizan la precisión con un sobrecoste computacional mínimo.

Por su parte ha sido evaluado en los conjuntos de datos KITTI, NYU y SUN RGB-D, BinsFormer, lo cual logra un rendimiento superior al estado del arte, con mejoras significativas en métricas como error relativo absoluto y precisión bajo umbrales. Los estudios de ablación confirman la efectividad de sus componentes, mientras que su capacidad de generalización se demuestra en evaluaciones cruzadas sin ajuste fino.

Fórmula:

Es la que define la predicción de profundidad final como una combinación lineal de las distribuciones de probabilidad por píxel y los centros de los intervalos (bins) adaptativos. Esta ecuación encapsula el núcleo del enfoque de clasificación-regresión, que combina representaciones probabilísticas con intervalos adaptativos generados por un decodificador Transformer para estimar mapas de profundidad continuos de alta precisión.

$$\hat{d} = \sum_{i=1}^{N} c(b_i) p_i$$

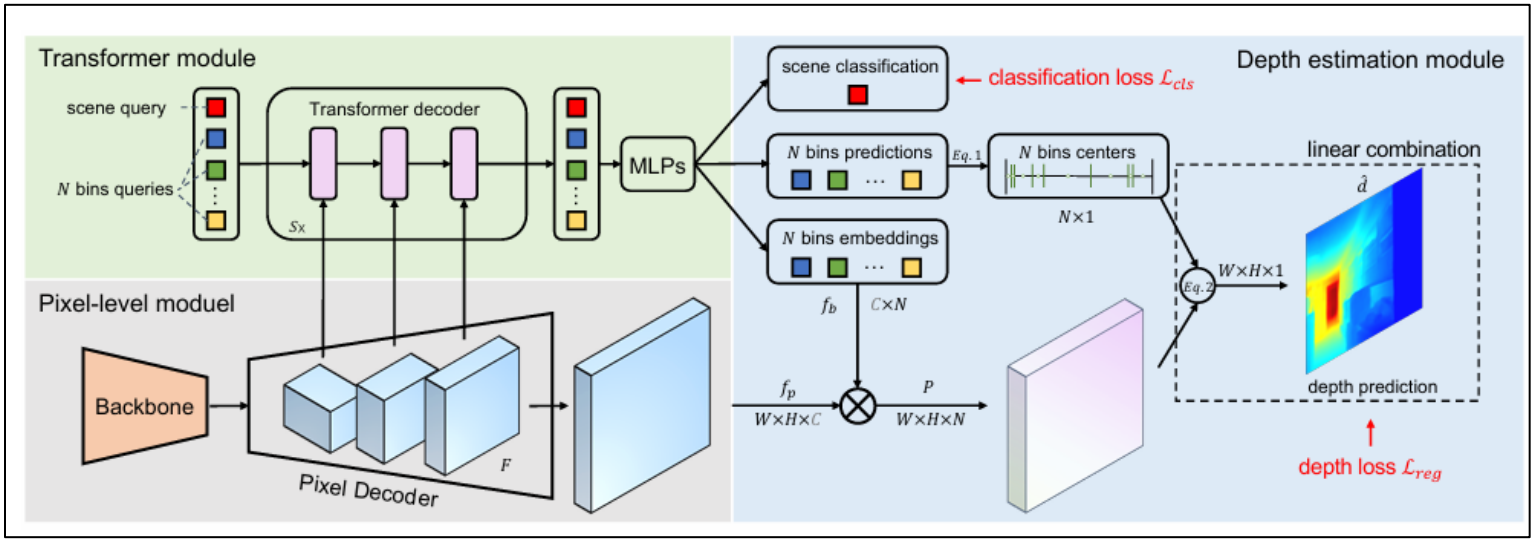
Por lo tanto, también se tiene la fórmula de la pérdida total utilizada para optimizar el modelo BinsFormer, que combina la pérdida de regresión para la estimación de profundidad y la pérdida de clasificación auxiliar para la tarea de comprensión de escenas:

$$\mathcal{L}_{total} = \sum_{s=1}^{S} \left(w_s \sum_{l=1}^{L} \left(\mathcal{L}_{reg}^{s,l} + \mu \mathcal{L}_{cls}^{s,l} \right) \right)$$

Otra fórmula clave es la del cálculo de los centros de los bins, en donde, los centros de los bins adaptativos se calculan a partir de las longitudes de bins predichas como:

$$c(b_i) = d_{min} + (d_{max} - d_{min}) \left(\frac{b_i}{2} + \sum_{j=1}^{i-1} b_j \right)$$

Esquema:


Transformer module
scene query
N bins queries
Transformer decoder
Sx
MLPs
Pixel-level moduel
Backbone
Pixel Decoder
F
scene classification
classification loss $\mathcal{L}_{cls}$
Depth estimation module
N bins predictions
Eq. 1
N bins centers
N×1
linear combination
N bins embeddings
f_b
C×N
f_p
W×H×C
P
W×H×N
Eq. 2
W×H×1
$\hat{d}$
depth prediction
depth loss $\mathcal{L}_{reg}$

Capítulo 18

“Trap Attention: Monocular Depth Estimation with Manual Traps”

Se basa en un enfoque innovador para la estimación de profundidad monocular, abordando la complejidad computacional de los métodos basados en atención multi-cabeza (MHA). Propone un mecanismo de atención eficiente, denominado trap attention (TA), que reduce la complejidad cuadrática de MHA $\partial(h^2w^2)$a lineal $\partial(hw)$ mediante una convolución profundidad-sabia de 7x7 para capturar información de largo alcance y trampas manuales para filtrar características relevantes. Estas trampas, definidas por cuatro funciones sinusoidales, clasifican píxeles en un espacio extendido, priorizando características informativas como bordes.

El modelo, construido como una red codificador-decodificador, utiliza un Vision Transformer como codificador y un decodificador basado en bloques de trampa (trap blocks) que integran TA, convoluciones profundidad-sabias y un MLP convolucional. Una unidad de selección de bloques (BS) mejora la representación de fondo al seleccionar píxeles máximos de bloques codificadores. La fusión de características se optimiza con interpolación de trampa no lineal, superando métodos tradicionales.

A su vez este modelo ha sido evaluado en **NYU, KITTI y SUN RGB-D,** el modelo (Trap-S, Trap-M, Trap-L) supera al estado del arte con hasta un 65% menos de parámetros. En NYU, Trap-L logra un Abs Rel de 0.094 y RMSE de 0.329; en KITTI, reduce el RMSE a 1.869. La evaluación cruzada en SUN RGB-D muestra una generalización robusta, con un Abs Rel de 0.141. Estudios de ablación confirman la eficacia de TA, interpolación de trampa y BS.

Fórmula:

Tenemos la pérdida de regresión invariante a escala utilizada para supervisar el entrenamiento del modelo de estimación de profundidad monoculares

$$\mathcal{L} = \alpha\sqrt{\frac{1}{N}\sum\left(\log\hat{d}_i - \log\hat{d}_i\right)^2 - \frac{\lambda}{N^2}\left(\sum\log\hat{d}_i - \log\hat{d}_i\right)^2}$$

Así mismo hay otras fórmulas.

Trap Attention (TA): La operación de atención de trampa (Trap Attention) se define como:

$$TA(X) = W_d * TI\big(PR(X)\big) + b_d$$

Trap Interpolation (TI): La interpolación de trampa aplica cuatro funciones de trampa manuales para clasificar características en un espacio extendido 2×2:

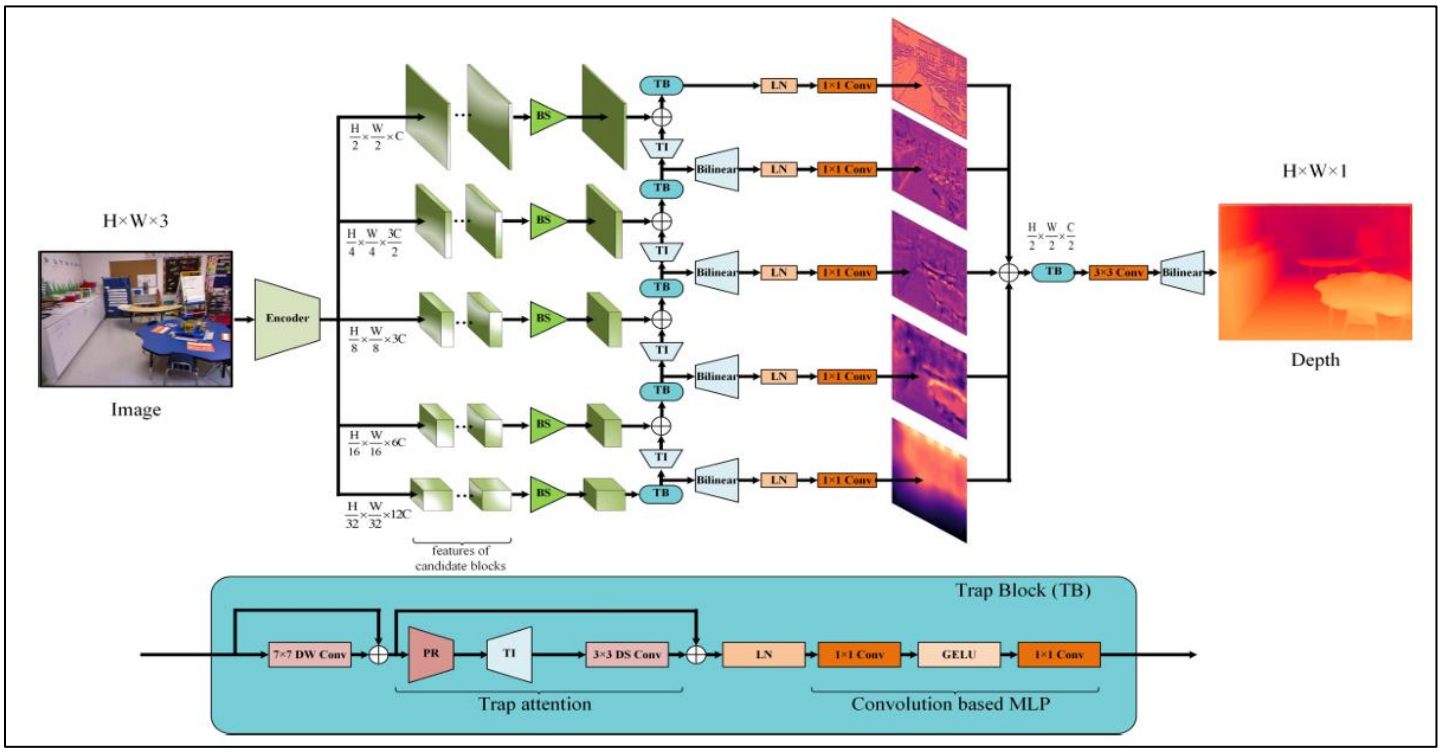
$$TI(x_i) = \begin{pmatrix} x_i * t_1(x_i), & x_i * t_2(x_i) \\ x_i * t_3(x_i), & x_i * t_4(x_i) \end{pmatrix}$$

Block Selection (BS): La unidad de selección de bloques selecciona el valor máximo píxel por píxel de n bloques candidatos del codificador Transformer:

$$Y_j = Max\big(B_j^1, B_j^2, \dots, B_j^k, \dots, B_j^n\big)$$

$$Chan(X) = W_{c2} * \big(GELU(W_{c1} * X + b_{c1})\big) + b_{c2}$$

Esquema:


H×W×3
Image
Encoder
BS
TB
TI
Bilinear
LN
1×1 Conv
features of candidate blocks
H×W×1
Depth
3×3 Conv
Trap Block (TB)
7×7 DW Conv
PR
3×3 DS Conv
GELU
Trap attention
Convolution based MLP

Capítulo 19

"Attention Attention Everywhere: Monocular Depth Prediction with Skip Attention"

Propone PixelFormer, una arquitectura innovadora para la estimación de profundidad monocular (MDE) que plantea el problema como un refinamiento de consultas de píxeles. Utiliza un codificador basado en Swin Transformer para extraer mapas de características multi-escala y un decodificador que refina consultas de píxeles iniciales desde la resolución más gruesa ($\frac{1}{32}$) hasta resoluciones más finas mediante el Skip Attention Module (SAM).

SAM emplea atención cruzada basada en ventanas para fusionar eficazmente características globales del decodificador con detalles locales del codificador, superando las limitaciones de las conexiones de salto basadas en convoluciones.

El modelo formula MDE como una tarea de clasificación-regresión, prediciendo profundidades mediante una combinación lineal de centros de bins adaptativos por imagen. El Bin Center Predictor (BCP) predice estos bins desde consultas de píxeles iniciales, integrando información de profundidad mediante supervisión directa con la verdad de terreno, lo que mejora la eficiencia frente a métodos previos como AdaBins. El Pixel Query Initialiser (PQI) genera consultas iniciales agregando información global con pyramid spatial pooling.

Ha sido evaluado en **NYUV2 y KITTI**, respectivamente, con mejoras en **RMSE** y precisión. En SUNRGB-D, sin ajuste fino, logra un 9.4% de mejora en Abs Rel (0.144), demostrando una robusta generalización. Los estudios de ablación confirman la eficacia de SAM y BCP, destacando la superioridad de la fusión basada en atención.

Fórmula:

Son las que describen la predicción de la profundidad d_i en un píxel i como una combinación lineal de los centros de los intervalos (bin centers) ponderados por las probabilidades predichas.

$$c(b_i) = d_{min} + (d_{max} - d_{min})\left(\frac{b_i}{2} + \sum_{j=1}^{i-1} b_j\right), i \in \{1, \ldots, n_{bins}\}$$

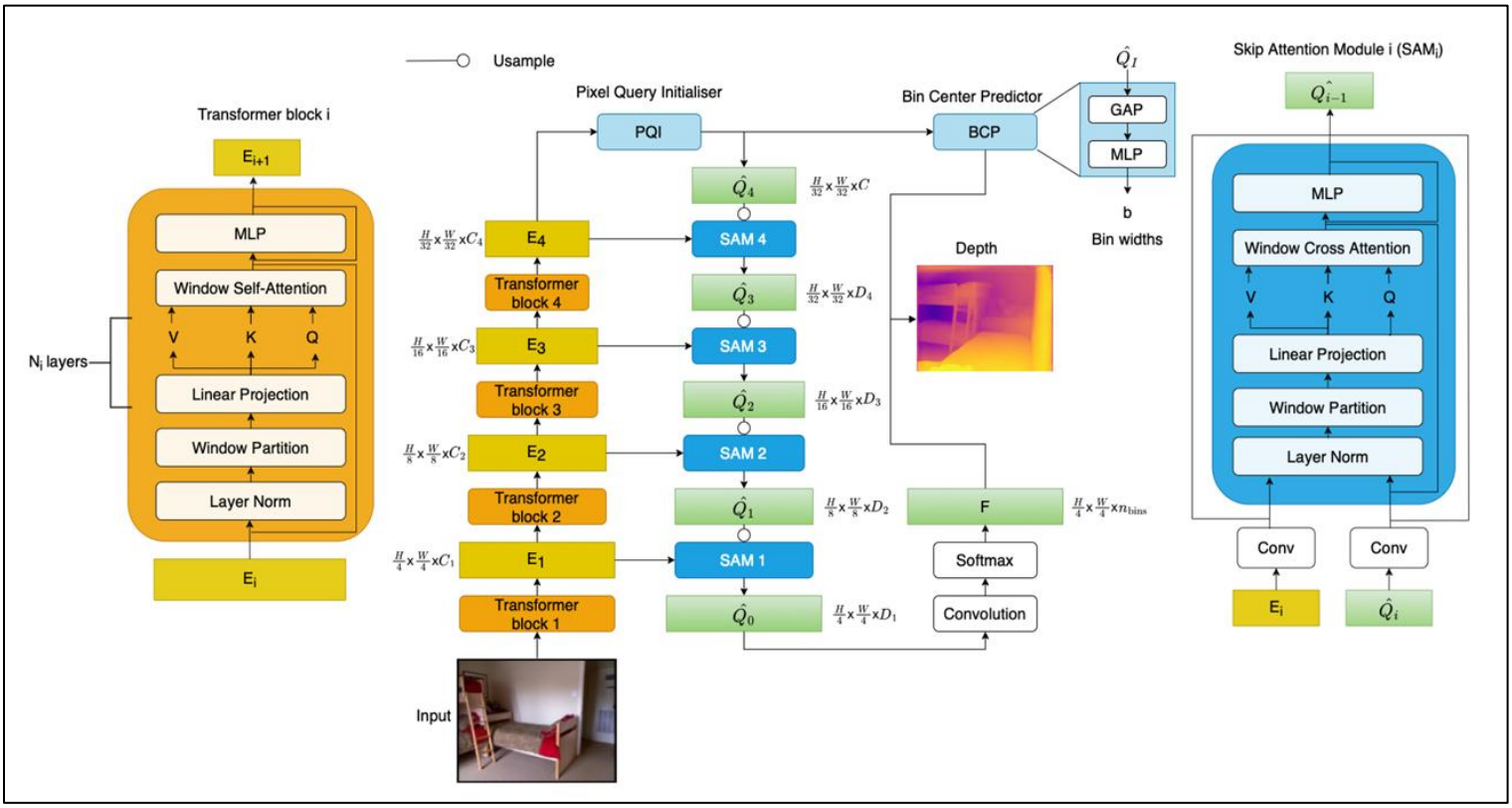
Esta fórmula encapsula la esencia del enfoque del paper, que modela la estimación de profundidad monocul ar (MDE) como un problema de clasificación-regresión.

$$d_i = \sum_{k=1}^{n_{bins}} c(b_k) p_{ik}$$

La función de pérdida principal utilizada para supervisar el entrenamiento del modelo PixelFormer. Se define como:

$$\mathcal{L}_{SILog} = \alpha \sqrt{\frac{1}{n}\sum_i g_i^2 - \frac{\lambda}{n^2}\left(\sum_i g_i\right)^2}$$

Esquema:


Usample
Transformer block i
E_{i+1}
MLP
Window Self-Attention
V
K
Q
N_i layers
Linear Projection
Window Partition
Layer Norm
E_i
Pixel Query Initialiser
PQI
Bin Center Predictor
BCP
$\hat{Q}_I$
GAP
MLP
b
Bin widths
$\hat{Q}_4$
$\frac{H}{32} \times \frac{W}{32} \times C$
$\frac{H}{32} \times \frac{W}{32} \times C_4$
E_4
SAM 4
Transformer block 4
$\hat{Q}_3$
$\frac{H}{32} \times \frac{W}{32} \times D_4$
Depth
$\frac{H}{16} \times \frac{W}{16} \times C_3$
E_3
SAM 3
Transformer block 3
$\hat{Q}_2$
$\frac{H}{16} \times \frac{W}{16} \times D_3$
$\frac{H}{8} \times \frac{W}{8} \times C_2$
E_2
SAM 2
Transformer block 2
$\hat{Q}_1$
$\frac{H}{8} \times \frac{W}{8} \times D_2$
F
$\frac{H}{4} \times \frac{W}{4} \times n_{\text{bins}}$
$\frac{H}{4} \times \frac{W}{4} \times C_1$
E_1
SAM 1
Softmax
Transformer block 1
$\hat{Q}_0$
$\frac{H}{4} \times \frac{W}{4} \times D_1$
Convolution
Input
Skip Attention Module i (SAM_i)
$\hat{Q}_{i-1}$
MLP
Window Cross Attention
V
K
Q
Linear Projection
Window Partition
Layer Norm
Conv
Conv
E_i
$\hat{Q}_i$

Capítulo 20

“Depth-Relative Self Attention for Monocular Depth Estimation”

Presenta **RElative Depth Transformer (RED-T),** un modelo innovador para la estimación de profundidad monocular (MDE) que aborda el problema de las "visual pits" (señales RGB engañosas como patrones o reflejos) que afectan la precisión. RED-T utiliza un Swin Transformer como backbone para extraer características multi-escala, un neck que agrega estas características en paralelo para preservar información global, y un head con atención relativa a la profundidad (depth-relative attention). Este mecanismo asigna mayores pesos de atención a píxeles de profundidad similar, reduciendo la dependencia de información RGB engañosa.

El modelo genera mapas de profundidad intermedios iterativamente, discretizando profundidades y calculando diferencias relativas para ajustar los pesos de atención. Esto fomenta que las características de píxeles con profundidades cercanas sean similares, mejorando la robustez ante visual pits. RED-T se entrena con pérdida invariante a escala y se evalúa en NYU-v2 y KITTI, superando al estado del arte.

Además, introduce un nuevo escenario de MDE con rango restringido, limitando las etiquetas de profundidad en entrenamiento (hasta 60 m en KITTI) para evaluar la generalización a profundidades no vistas. RED-T muestra menor degradación comparado con AdaBins. Ablaciones confirman la eficacia del depth-relative bias. Con 248M de parámetros, RED-T es competitivo, pero requiere más tiempo de inferencia.

Fórmula:

Se tiene la ecuación de la atención relativa a la profundidad (depth-relative self-attention), que ajusta los pesos de atención basándose en la diferencia de profundidad relativa entre píxeles.

$$A_h(Q_h, K_h, B_h) = Softmax\left(\frac{Q_h K_h^T}{\sqrt{d_h}} + B_h\right)$$

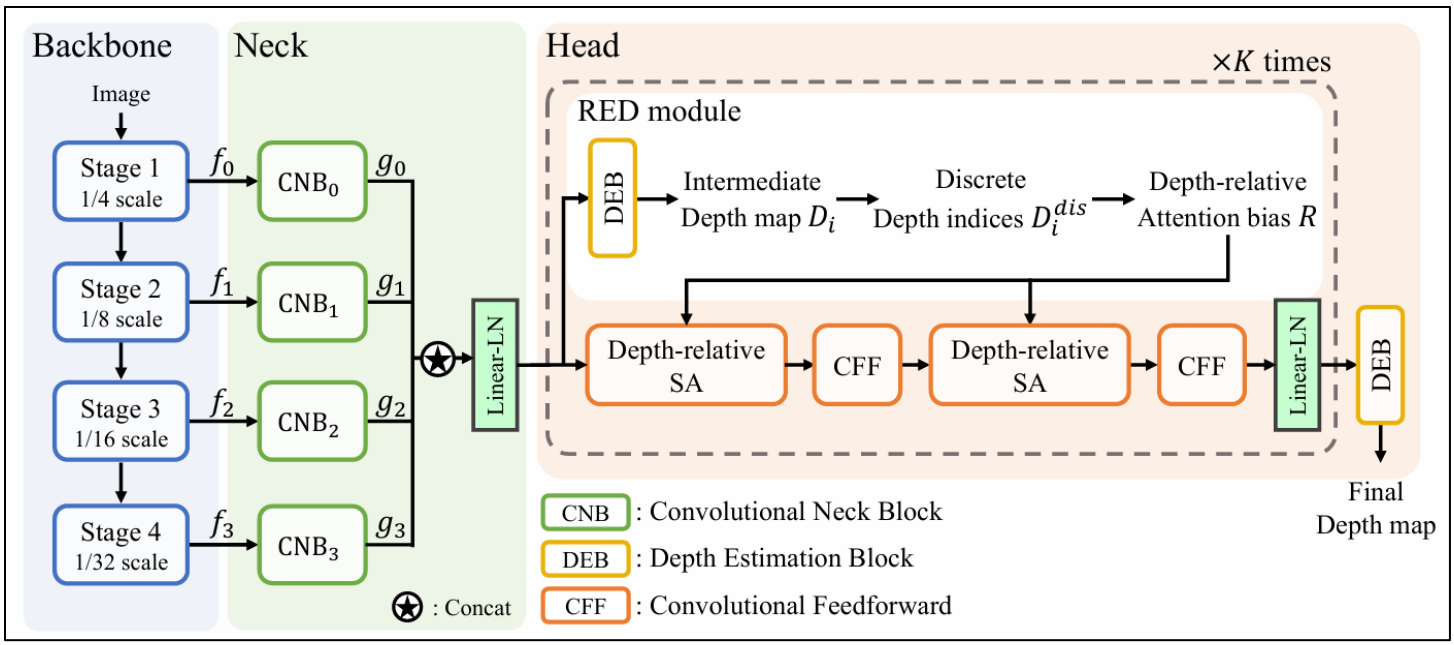
Obteniendo como resultado:

$$SA_h(Q_h, K_h, V_h, B_h) = A_h(Q_h, K_h, B_h)V_h$$

Otra de las fórmulas es la pérdida total utilizada para supervisar el entrenamiento del modelo RED-T es una versión promediada de la pérdida invariante a escala (Scale-Invariant Loss) aplicada sobre todos los mapas de profundidad intermedios.

$$L_i = \alpha \sqrt{\frac{1}{T}\sum_{x=0}^{T-1} h_{i,x}^2 - \frac{\lambda}{T}\left(\sum_{x=0}^{T-1} h_{i,x}\right)^2} \;;\; L_{total} = \sum_{i=0}^{K} \frac{L_i}{K+1}$$

Esquema:


Backbone
Neck
Head
Image
Stage 1
1/4 scale
Stage 2
1/8 scale
Stage 3
1/16 scale
Stage 4
1/32 scale
f_0
f_1
f_2
f_3
CNB_0
CNB_1
CNB_2
CNB_3
g_0
g_1
g_2
g_3
: Concat
Linear-LN
$\times K$ times
RED module
DEB
Intermediate
Depth map D_i
Discrete
Depth indices D_i^{dis}
Depth-relative
Attention bias R
Depth-relative
SA
CFF
Depth-relative
SA
CFF
Linear-LN
DEB
Final
Depth map
CNB : Convolutional Neck Block
DEB : Depth Estimation Block
CFF : Convolutional Feedforward

Capítulo 21

“NeW CRFs: Neural Window Fully-connected CRFs for Monocular Depth Estimation”

El artículo presenta un método innovador para estimar mapas de profundidad a partir de imágenes monoculares, abordando el problema de la ambigüedad inherente mediante la optimización de Campos Aleatorios Condicionales (CRFs) completamente conectados. Para reducir la complejidad computacional de los FC-CRFs, los autores dividen la imagen en ventanas y aplican CRFs dentro de cada una, haciendo factible su uso. Incorporan un mecanismo de atención multi-cabeza para capturar relaciones entre nodos, integrando este módulo de CRFs neurales como decodificador en una red de estructura bottom-up-top-down, con un transformador visual como codificador.

El método mejora significativamente el rendimiento en los conjuntos de datos KITTI, NYUv2 y MatterPort3D, superando a enfoques previos. En KITTI, reduce el error Abs-Rel en un 10.3% y el RMSE en un 9.8%; en NYUv2, el Abs-Rel cae un 7.8% y el RMSE un 8.2%. También establece un nuevo estándar en imágenes panorámicas en MatterPort3D, incluso sin adaptaciones específicas, mostrando robustez en diversos escenarios.

Esta red utiliza una estrategia de ventanas desplazadas para conectar información entre ventanas y un módulo de agrupación piramidal (PPM) para agregar contexto global. La optimización se realiza con pérdida SILog, y el modelo se entrena end-to-end en PyTorch. Además, visualizaciones de nubes de puntos demuestran la capacidad del modelo para reconstruir estructuras 3D precisas, incluso en imágenes panorámicas no vistas, destacando su potencial para aplicaciones reales como la reconstrucción 3D y la robótica.

Fórmula:

La fórmula más fundamental del artículo es la ecuación de la energía de los Campos Aleatorios Condicionales totalmente conectados (FC-CRFs).

$$E(x) = \sum_i \psi_u(x_i) + \sum_{ij} \psi_p(x_i, x_j)$$

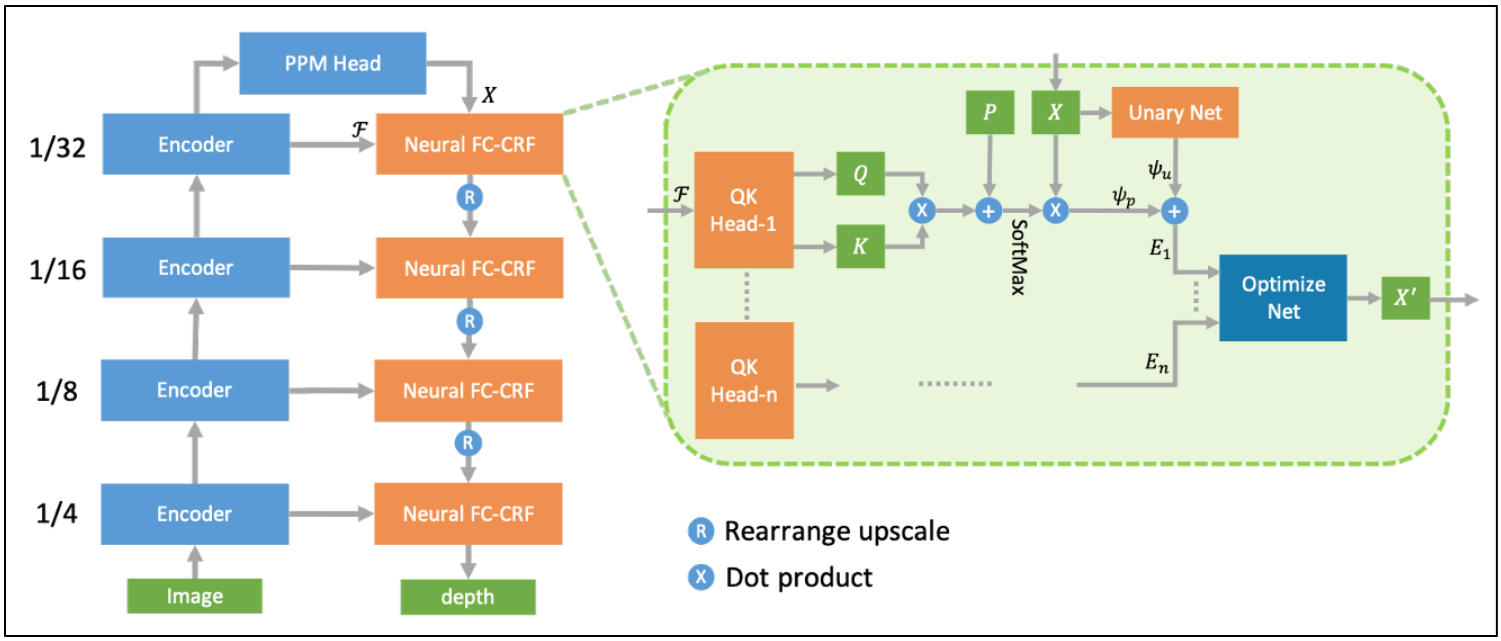
Además, la función potencial por pares ψp conecta pares de nodos como:

$$\psi_p = \mu(x_i, x_j) f(x_i, x_j) g(I, I_j) h(p_i, p_j)$$

Finalmente, la pérdida utilizada para supervisar el entrenamiento es la pérdida logarítmica invariante a escala (Scale-Invariant Logarithmic Loss, SILog), que se define como:

$$\mathcal{L} = \alpha \sqrt{\frac{1}{K}\sum_i \triangle\, d_i^2 - \frac{\lambda}{K^2}\left(\sum_i \Delta d_i\right)^2}$$

Esquema:


PPM Head
X
1/32
Encoder
$\mathcal{F}$
Neural FC-CRF
R
1/16
Encoder
Neural FC-CRF
R
1/8
Encoder
Neural FC-CRF
R
1/4
Encoder
Neural FC-CRF
Image
depth
$\mathcal{F}$
QK Head-1
QK Head-n
Q
K
X
P
X
Unary Net
SoftMax
ψ_u
ψ_p
E_1
E_n
Optimize Net
X'
R Rearrange upscale
X Dot product

Capítulo 22

"DepthFormer: Exploiting Long-range Correlation and Local Information for Accurate Monocular Depth Estimation"

Propone un marco innovador para la estimación de profundidad monocular supervisada. El método aborda las limitaciones de los enfoques basados en CNN y Vision Transformer (ViT) mediante un codificador paralelo que combina una rama Transformer, experta en correlaciones de largo alcance, y una rama convolucional, que preserva información local. Un estudio piloto demuestra que los Transformers superan a las CNN en objetos distantes, pero fallan en estimaciones cercanas debido a la falta de sesgo inductivo espacial.

Para superar la fusión insuficiente de características, se introduce el módulo de Agregación Jerárquica e Interacción Heterogénea (HAHI), que mejora las características del Transformer mediante autoatención deformable y modela la afinidad entre características heterogéneas (Transformer y CNN) en un enfoque de traducción conjunto a conjunto. El uso de un esquema deformable reduce el costo de memoria de la atención global en mapas de alta resolución.

DepthFormer logra un rendimiento superior en los conjuntos de datos **KITTI, NYU y SUN RGB-D**, superando a métodos previos como AdaBins, DPT y NeWCRFs. Su generalización se evidencia en SUN RGB-D sin ajuste fino. La arquitectura es escalable, compatible con variantes de Transformer como Swin, y reduce el tiempo de entrenamiento al optimizar la rama convolucional. Además, DepthFormer tiene aplicaciones en conducción autónoma y reconstrucción 3D, con potencial para futuras mejoras en atención dedicada y fusión multimodal.

Fórmula:

Se tiene la que tiene relación con el módulo Hierarchical Aggregation and Heterogeneous Interaction (HAHI), específicamente la ecuación que describe el Deformable Attention (DAttn) utilizado para la mejora de características y la modelación de afinidad entre características heterogéneas:

$$DAttn(x_q, x_v, p_q) = \sum_{k \in \Omega_k} A_{qk} x_v (p_q + \Delta p_{qk})$$

Esta fórmula es central para el módulo HAHI, que mejora las características del Transformer y modela la afinidad entre las características del Transformer (F) y las de la rama convolucional (G).

Asi mismo, la rama Transformer del encoder de DepthFormer utiliza capas Transformer para extraer características que modelan correlaciones de largo alcance.

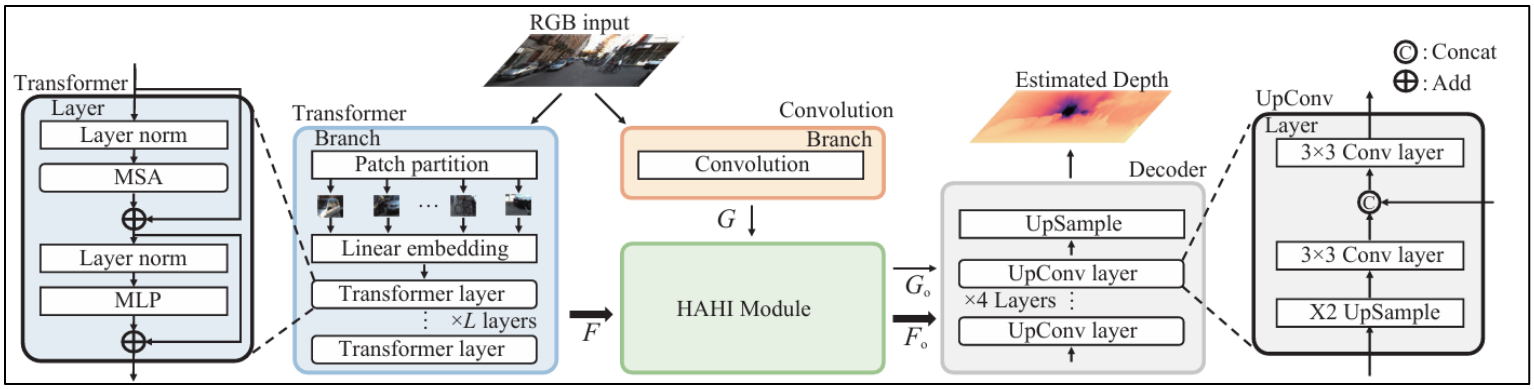
$$\hat{z}^l = MSA\left(LN\left(z^{l-1}\right)\right) + z^{l-1}$$

$$z^l = MLP\left(LN\left(\hat{z}^l\right)\right) + \hat{z}^l$$

Finalmente, la pérdida utilizada para entrenar DepthFormer es la pérdida logarítmica invariante a escala (Scale-Invariant Logarithmic Loss, SILog), definida como:

$$\mathcal{L}_{pixel} = \alpha \sqrt{\frac{1}{T}\sum_i h_i^2 - \frac{\lambda}{T^2}\left(\sum_i h_i\right)^2}$$

Esquema:


RGB input
Transformer
Layer
Layer norm
MSA
Layer norm
MLP
Transformer
Branch
Patch partition
Linear embedding
Transformer layer
×L layers
Transformer layer
Convolution
Branch
Convolution
G
F
HAHI Module
G_o
F_o
Estimated Depth
Decoder
UpSample
UpConv layer
×4 Layers
UpConv layer
UpConv
Layer
3×3 Conv layer
3×3 Conv layer
X2 UpSample
: Concat
: Add

Capítulo 23

“ViP-DeepLab: Learning Visual Perception with Depth-aware Video Panoptic Segmentation”

El artículo presenta ViP-DeepLab, un modelo unificado que aborda el problema de proyección inversa en visión, formulado como Depth-aware Video Panoptic Segmentation (DVPS). Esta combina la estimación de profundidad monocular y la segmentación panóptica de video, prediciendo ubicación espacial, clase semántica e identificación de instancias consistente en el tiempo para cada punto 3D a partir de secuencias de imágenes 2D.

ViP-DeepLab extiende Panoptic-DeepLab con una cabeza de predicción de profundidad y una rama para instancias en el siguiente fotograma, utilizando regresión de centros para seguimiento de objetos en dos fotogramas consecutivos. Introduce Cascade-ASPP para ampliar el campo receptivo y un método de costura basado en IoU para propagar identificadores de instancias. La estimación de profundidad se modela como regresión densa, optimizada con una combinación de errores logarítmicos y relativos.

Se proponen dos conjuntos de datos derivados, Cityscapes-DVPS y SemKITTI-DVPS, con anotaciones de profundidad y segmentación panóptica, y una nueva métrica, Depth-aware Video Panoptic Quality (DVPQ), que evalúa precisión en segmentación y profundidad. ViP-DeepLab logra resultados sobresalientes, lidera el ranking en KITTI MOTS para peatones y en estimación de profundidad monocular KITTI. Ablaciones muestran que preentrenamiento, módulos contextuales y aumentos mejoran el rendimiento.

El modelo destaca en consistencia temporal y precisión, con aplicaciones en conducción autónoma y reconstrucción 3D. Los conjuntos de datos y códigos están disponibles públicamente, estableciendo un referente para investigaciones futuras en DVPS.

Fórmula:

La métrica DVPQ es fundamental para evaluar el desempeño en la tarea de DVPS, ya que combina la calidad de la segmentación panóptica de video (VPQ) con la precisión de la estimación de profundidad.

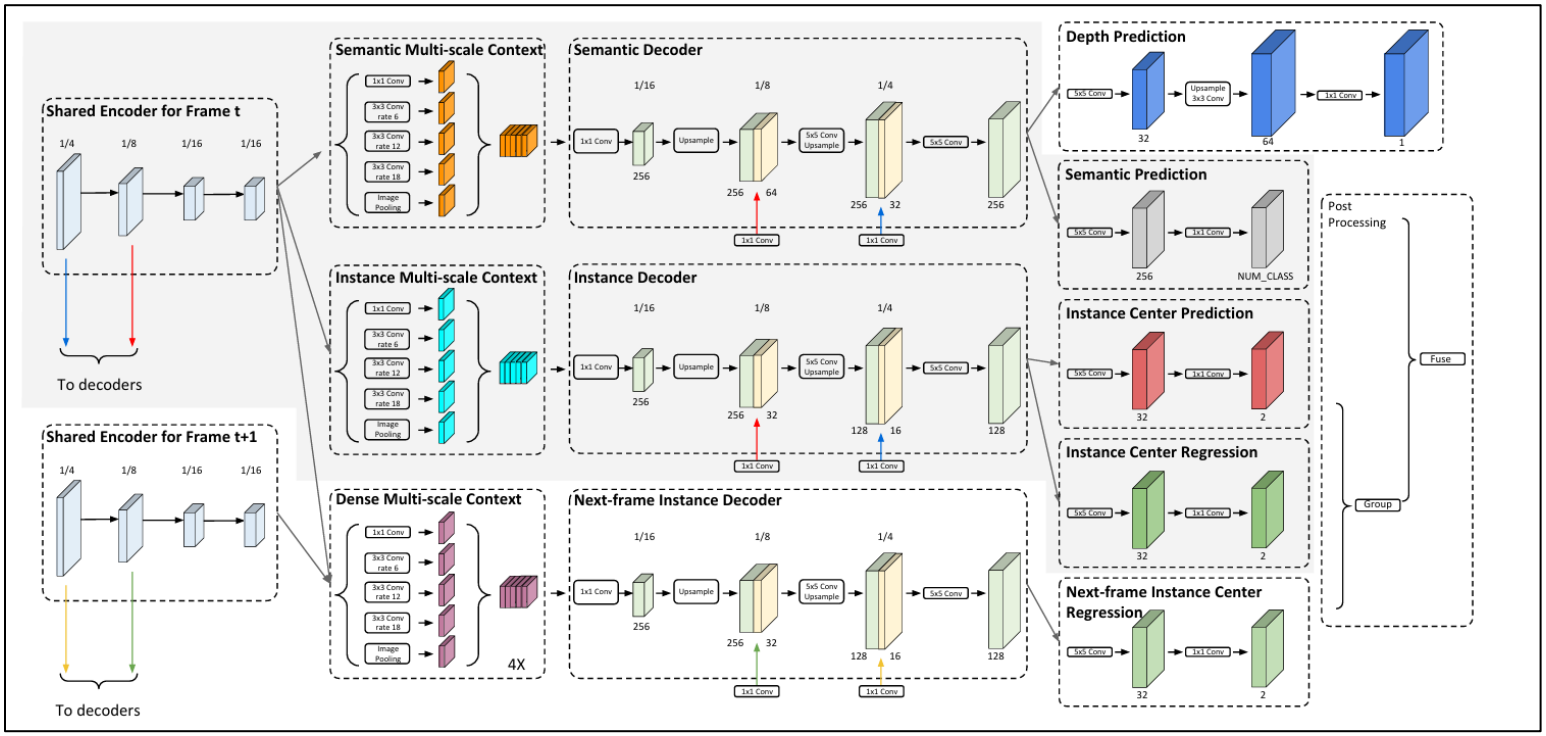
$$PQ\left(\left[\Vert_{i=t}^{t+k-1}\left(\hat{P}_i^c, P_i^{id}\right), \Vert_{i=t}^{t+k-1}\left(Q_i^c, Q_i^{id}\right)\right]_{t=1}^{T-k+1}\right)$$

La estimación de profundidad monoculada se modela como un problema de regresión densa, y la función de pérdida utilizada para entrenar el modelo es una combinación del error logarítmico invariante a escala y el error cuadrático relativo.

$$\mathcal{L}_{depth}(d,\hat{d}) = \frac{1}{n}\sum_i\left(\log d_i - \log \hat{d}_i\right)^2 - \frac{1}{n^2}\left(\sum_i\left(\log d_i - \log \hat{d}_i\right)^2 + \left(\frac{1}{n}\sum_i\left(\frac{d_i - \hat{d}_i}{d_i}\right)^2\right)^{0.5}\right)$$

Esta pérdida combina el error logarítmico invariante a escala y el error cuadrático relativo para optimizar la predicción de profundidad, contribuyendo al éxito de ViP-DeepLab en el benchmark de KITTI.

Esquema:


Shared Encoder for Frame t
1/4
1/8
1/16
1/16
To decoders
Shared Encoder for Frame t+1
1/4
1/8
1/16
1/16
To decoders
Semantic Multi-scale Context
1x1 Conv
3x3 Conv rate 6
3x3 Conv rate 12
3x3 Conv rate 18
Image Pooling
Instance Multi-scale Context
1x1 Conv
3x3 Conv rate 6
3x3 Conv rate 12
3x3 Conv rate 18
Image Pooling
Dense Multi-scale Context
1x1 Conv
3x3 Conv rate 6
3x3 Conv rate 12
3x3 Conv rate 18
Image Pooling
4X
Semantic Decoder
1/16
1/8
1/4
1x1 Conv
256
Upsample
256
64
5x5 Conv Upsample
256
32
5x5 Conv
256
1x1 Conv
1x1 Conv
Instance Decoder
1/16
1/8
1/4
1x1 Conv
256
Upsample
256
32
5x5 Conv Upsample
128
16
5x5 Conv
128
1x1 Conv
1x1 Conv
Next-frame Instance Decoder
1/16
1/8
1/4
1x1 Conv
256
Upsample
256
32
5x5 Conv Upsample
128
16
5x5 Conv
128
1x1 Conv
1x1 Conv
Depth Prediction
5x5 Conv
32
Upsample 3x3 Conv
64
1x1 Conv
1
Semantic Prediction
5x5 Conv
256
1x1 Conv
NUM_CLASS
Instance Center Prediction
5x5 Conv
32
1x1 Conv
2
Instance Center Regression
5x5 Conv
32
1x1 Conv
2
Next-frame Instance Center Regression
5x5 Conv
32
1x1 Conv
2
Post Processing
Fuse
Group

Capítulo 24

“SideRT: A Real-time Pure Transformer Architecture for Single Image Depth Estimation”

El artículo presenta SideRT, como una arquitectura de transformadores pura para la estimación de profundidad a partir de una sola imagen (SIDE) en tiempo real. A diferencia de métodos previos basados en CNN o combinaciones con transformadores, SideRT elimina convoluciones, logrando un equilibrio entre precisión y velocidad. Utiliza un codificador-decodificador con Swin Transformers como backbone y un decodificador ligero con módulos de Cross-Scale Attention (CSA) y Multi-Scale Refinement (MSR). CSA fusiona características de diferentes escalas según similitud semántica, capturando contexto global, mientras que MSR refina características según correspondencia espacial. Además, se implementa Multi-Stage Supervision (MSS) para facilitar el entrenamiento.

SideRT alcanza un rendimiento de vanguardia en los conjuntos de datos KITTI y NYU, mejorando el métrico AbsRel en un 6.9% (KITTI) y 9.7% (NYU). Este diseño eficiente, con pocos parámetros aprendibles, permite predicciones precisas en tiempo real, superando limitaciones de velocidad de modelos basados en atención. La visualización muestra que CSA amplía significativamente el campo receptivo, mejorando la predicción de profundidad en detalles finos.

El estudio destaca que una arquitectura de transformadores pura puede ser práctica para aplicaciones como conducción autónoma o realidad aumentada, ofreciendo un modelo ligero y efectivo. Los experimentos confirman la contribución de CSA, MSR y MSS al rendimiento, consolidando a SideRT como la primera red basada en transformadores que logra resultados de vanguardia en tiempo real para SIDE.

Fórmula:

La fórmula fundamental del módulo CSA, que permite la fusión de mapas de características de diferentes escalas basándose en la similitud semántica, se define como:

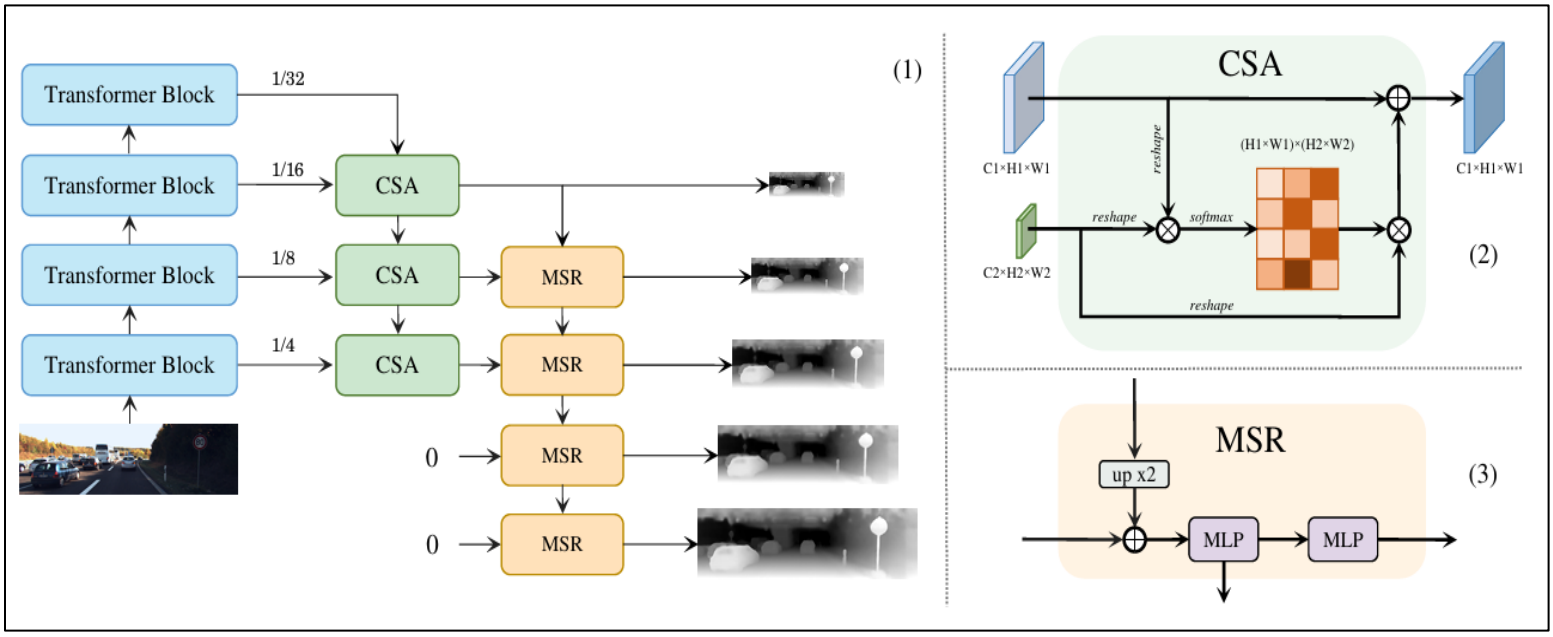
$$F_{12} = L(F_1) + softmax\big(L(F_1) * l(F_2)\big) * L(F_2)$$

La función de pérdida utilizada para entrenar el modelo en cada etapa de la supervisión multi-etapa (MSS) se basa en una pérdida de raíz cuadrada en el espacio logarítmico, definida como:

$$L(y, y^*) = \sqrt{\frac{1}{n}\sum_{i\epsilon V}^{N_v} d_i^2 - \frac{\lambda}{n^2}\left(\sum_{i\epsilon V}^{N_v} d_i\right)^2}$$

$$d_i = \log y_i - \log y_i^*$$

Esquema:


Transformer Block
Transformer Block
Transformer Block
Transformer Block
1/32
1/16
1/8
1/4
CSA
CSA
CSA
MSR
MSR
MSR
MSR
0
0
(1)
CSA
C1×H1×W1
C2×H2×W2
reshape
softmax
(H1×W1)×(H2×W2)
reshape
reshape
C1×H1×W1
(2)
MSR
up x2
MLP
MLP
(3)

Capítulo 25

“Patch-Wise Attention Network for Monocular Depth Estimation”

El artículo presenta una red de atención por parches (Patch-Wise Attention, PWA) para la estimación de profundidad monocular. Este método aborda las limitaciones de los enfoques basados en CNN al centrarse en las relaciones entre píxeles vecinos en áreas locales, mejorando la reconstrucción de objetos y bordes. La arquitectura, basada en un esquema codificador-decodificador, utiliza DenseASPP para capturar información multiescala y aplica PWA tras la última etapa de upsampling. PWA extrae parches de tamaño predefinido, aplicando secuencialmente atención por canal y espacial para generar mapas de atención que refinan características locales, integrando tanto al contexto global y local.

El método se evaluó en los conjuntos de datos KITTI y NYU Depth V2, superando a los enfoques de vanguardia. En KITTI Eigen split, el modelo basado en ResNeXt101 logró un AbsRel de 0.057 (0-50 m) y 0.060 (0-80 m), mientras que en el benchmark online de KITTI, obtuvo un SILog de 11.45, liderando la métrica principal. En NYU Depth V2, el modelo basado en DenseNet161 redujo AbsRel a 0.105 y RMSE a 0.404, destacando incluso con backbones ligeros como MobileNetV2. Los estudios de ablación confirmaron la eficacia de PWA frente a métodos de atención global como CBAM y DualAttention, mostrando mejoras significativas con un tamaño de parche adecuado.

La implementación, realizada en PyTorch con backbones preentrenados, utilizó optimización ADAM y aumento de datos. Los resultados cualitativos destacan bordes más nítidos y mejor reconstrucción de objetos.

Fórmula:

Las fórmulas más fundamentales del se encuentran en las secciones que describen el módulo de Patch-Wise Attention (PWA) y la función de pérdida utilizada para entrenar la red.

La fórmula para generar el mapa de atención de canal parche a parche, que refina las características de entrada basándose en la información de contexto local y global, se define como:

$$F_c = Conv_c([F_G; MaxPool_s(F); AvgPool_s(F)]) = Conv_c\left(\left[F_G; F_{max}^c; F_{avg}^c\right]\right)$$

$$E_i^c = \sigma\left(MLP_i(F_i^c)\right)$$

La fórmula para generar el mapa de atención espacial parche a parche, que refina aún más las características basándose en relaciones espaciales locales, se define como:

$$F_s = Conv_s([F'; F_G^I])$$

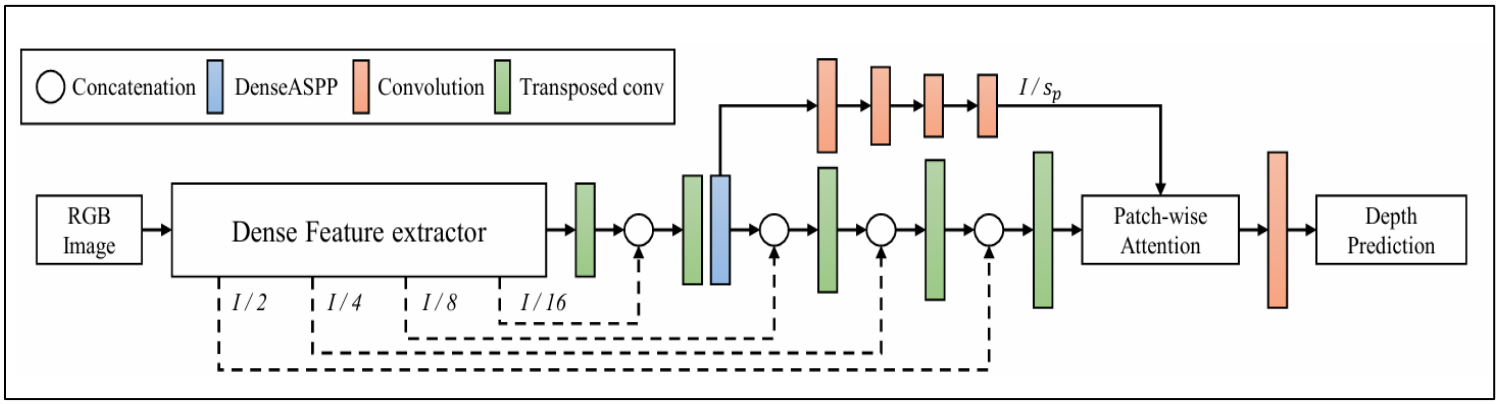
$$E_j^s = \sigma\left(Conv_j\left(\left[MaxPool_c\left(F_{s,j}\right); AvgPool_c\left(F_{s,j}\right)\right]\right)\right) = \sigma\left(Conv_j\left(\left[F_{max,j}^s; F_{avg,j}^s\right]\right)\right) = \sigma\left(Conv_j\left(F_j^s\right)\right)$$

La función de pérdida utilizada para entrenar la red, combina un error $l2$ por elemento y un error invariante a escala, definida como:

$$d_i = \log y_i - \log y_i^*$$

$$L = \sqrt{\frac{1}{n}\sum_i d_i^2 - \frac{\lambda}{n^2}\left(\sum_i d_i\right)^2}$$

Esquema:



Capítulo 26

“Bidirectional Attention Network for Monocular Depth Estimation”

El artículo presenta la Bidirectional Attention Network (BANet), una arquitectura de extremo a extremo para la estimación de profundidad monocular (MDE), desarrollada por investigadores de Huawei. BANet aborda las limitaciones de las CNN al integrar información local y global mediante módulos de atención bidireccional, inspirados en RNN bidireccionales para traducción automática. Utiliza un backbone DenseNet161, transformando mapas de características de cinco etapas. Estos mapas se procesan con atención forward (hacia etapas iniciales) y backward (hacia etapas finales), generando pesos de atención por etapa que refinan predicciones, incorporando contexto global vía pooling y upsampling.

Las cinco etapas de atención se describen así: Etapa 1 activa regiones genéricas; Etapa 2 enfoca el plano del suelo; Etapa 3 detecta objetos individuales (autos, peatones); Etapa 4 mide variaciones de profundidad intra-objeto; Etapa 5 captura el punto de fuga para estimar límites de profundidad. BANet-Full, la versión completa, logra un equilibrio entre precisión y eficiencia, con menos parámetros (35.64M) y tiempo de inferencia (31ms) que competidores como DORN (91.31M, 57ms).

Ha sido evaluada en KITTI y DIODE, en KITTI, alcanza un SILog de 11.67 en el leaderboard, destacando en regiones delicadas como troncos de árboles. En DIODE, mejora la detección de estructuras finas en interiores y exteriores. Los experimentos confirman que la atención backward es más crítica que la forward, y la agregación de contexto global reduce falsos negativos en escenarios complejos, haciendo a BANet ideal para aplicaciones sensibles como conducción autónoma.

Fórmula:

Las fórmulas más fundamentales del artículo se encuentran en la sección que describe los Bidirectional Attention Modules, específicamente en las ecuaciones que formalizan la generación de mapas de atención bidireccional y la predicción final del mapa de profundidad. Estas fórmulas son el núcleo de la arquitectura propuesta, BANet, que introduce un mecanismo de atención bidireccional para integrar información local y global en la estimación de profundidad monocular, se expresan como:

$$s_i^f = \mathcal{D}_i^f(s_i) \qquad s_i^b = \mathcal{D}_i^b(s_i)$$

$$a_i^f = \mathcal{A}_i^f\left(s_1^f, s_2^f, \dots, s_i^f\right) \qquad a_i^b = \mathcal{A}_i^b\left(s_i^b, s_{i+1}^b, \dots, s_N^b\right)$$

$$\mathcal{G}^f = a_1^f \otimes a_2^f \otimes \dots \otimes a_N^f \qquad \mathcal{G}^b = a_1^b \otimes a_2^b \otimes \dots \otimes a_N^b$$

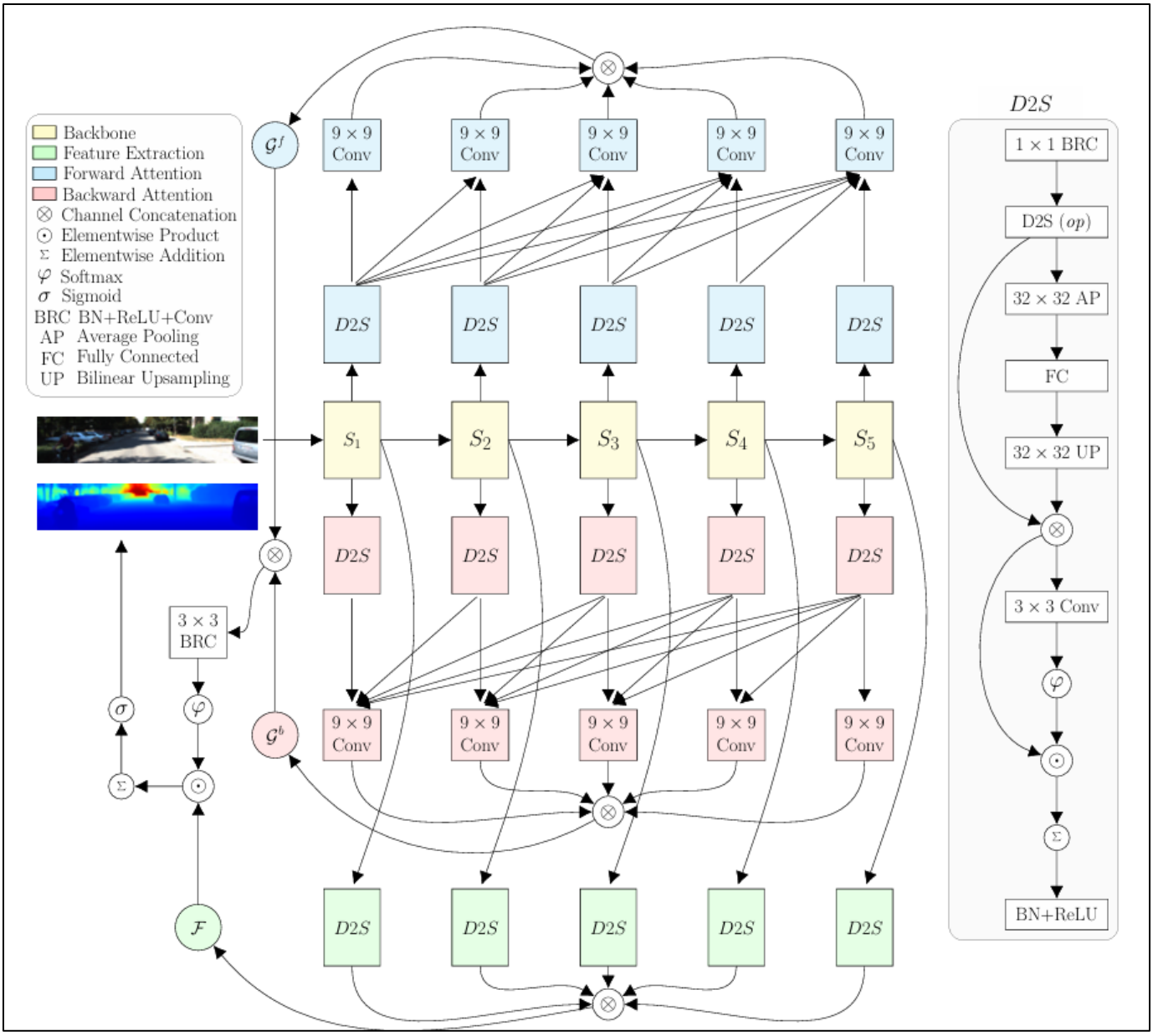
$$\mathcal{A} = \varphi\left(\mathcal{G}\left(\mathcal{G}^f \otimes \mathcal{G}^b\right)\right)$$

Las fórmulas que describen la generación de representaciones de características por etapa y la predicción final del mapa de profundidad se definen como:

$$f_i = \mathcal{D}_i^{\mathcal{F}}(s_i);\ \mathcal{F} = f_1 \otimes f_2 \otimes \dots \otimes f_N$$

$$\widehat{D}_u = \sum \mathcal{A} \odot \mathcal{F},\ \widehat{D} = \sigma\left(\widehat{D}_u\right)$$

Esquema:


Backbone
Feature Extraction
Forward Attention
Backward Attention
Channel Concatenation
Elementwise Product
Elementwise Addition
Softmax
Sigmoid
BRC BN+ReLU+Conv
AP Average Pooling
FC Fully Connected
UP Bilinear Upsampling
9 × 9 Conv
D2S
S_1
S_2
S_3
S_4
S_5
3 × 3 BRC
$\mathcal{G}^f$
$\mathcal{G}^b$
$\mathcal{F}$
D2S
1 × 1 BRC
D2S (op)
32 × 32 AP
FC
32 × 32 UP
3 × 3 Conv
BN+ReLU

Capítulo 27

“From Big to Small: Multi-Scale Local Planar Guidance for Monocular Depth Estimation”

El artículo presenta un método innovador para estimar la profundidad a partir de una sola imagen utilizando redes neuronales convolucionales profundas (DCNNs). Propone una arquitectura de red que incorpora capas de guía planar local (LPG) en múltiples etapas de la fase de decodificación, mejorando la precisión en la predicción de profundidad. Estas capas estiman coeficientes planares 4D para regiones locales, permitiendo una reconstrucción eficiente de la profundidad en resolución completa, a diferencia de métodos tradicionales que usan upsampling simple o conexiones de salto.

La red sigue un esquema de codificación-decodificación, utilizando redes como ResNet, DenseNet o ResNext como extractores de características densas, y emplea atrous spatial pyramid pooling (ASPP) para capturar contexto multiescala. Las capas LPG, ubicadas en resoluciones de 1/8, 1/4 y 1/2, guían las características hacia la predicción final, combinándolas de manera no lineal para lograr estimaciones precisas.

Evaluado en los conjuntos de datos NYU Depth V2 y KITTI, el método supera a los enfoques de vanguardia, mostrando mejoras significativas en métricas como **Abs Rel, RMSE** y umbrales de precisión. Un estudio de ablación valida la efectividad de las capas LPG, que añaden pocos parámetros, pero mejoran notablemente el rendimiento. Aunque destaca en bordes de objetos, se observan artefactos en regiones con datos de profundidad escasos, como el cielo en KITTI, lo que sugiere futuras mejoras con pérdidas de reconstrucción fotométrica. Este modelo ofrece una solución robusta para aplicaciones como la robótica y conducción autónoma, con resultados cualitativos y cuantitativos superiores en entornos interiores y exteriores.

Fórmula:

La fórmula clave para la capa de guía planar local (LPG) describe cómo se estiman los coeficientes de un plano 4D para cada celda espacial en un mapa de características de resolución y cómo estos coeficientes se utilizan para reconstruir estimaciones de profundidad relativa en la resolución completa H×W.

$$\tilde{c}^{k*k} = LPG(F_k) \in \mathbb{R}^{H|k*W|k*4}$$

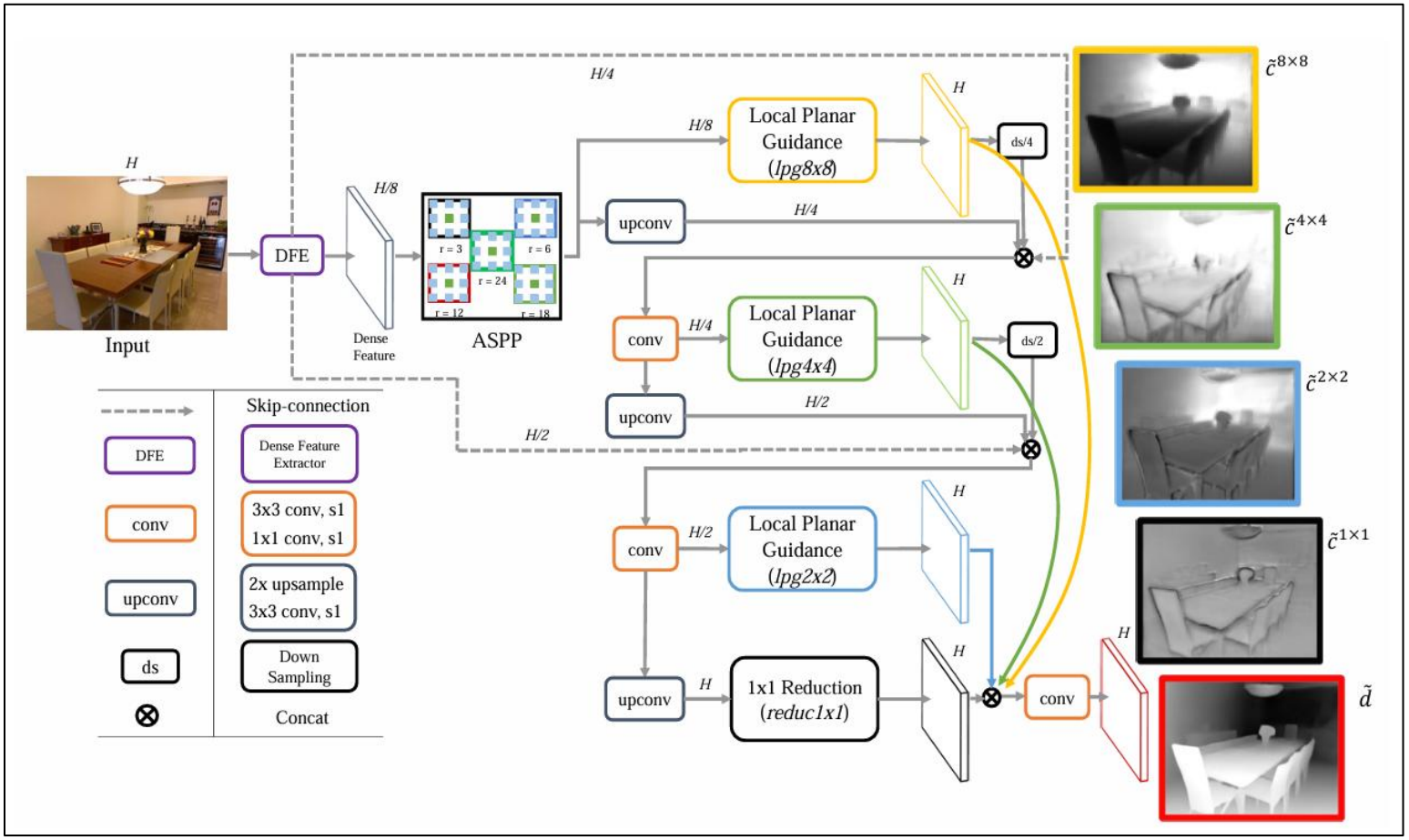
Para la perdida base por su parte se tiene las siguientes formulas:

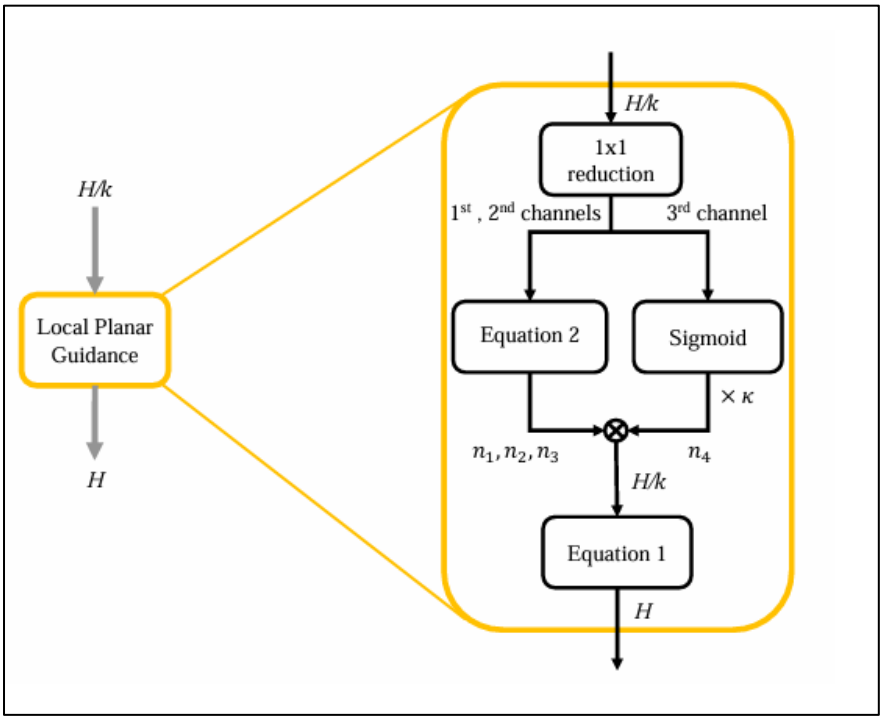
$$D(g) = \frac{1}{T}\sum_i g_i^2 - \frac{\lambda}{T^2}\left(\sum_i g_i\right)^2$$

$$D(g) = \frac{1}{T}\sum_i g_i^2 - \left(\frac{1}{T}\sum_i g_i\right)^2 + (1-\lambda)\left(\frac{1}{T}\sum_i g_i\right)^2$$

$$L = \propto \sqrt{D(g)}$$

Esquema:


H
Input
DFE
H/8
Dense Feature
r = 3
r = 6
r = 24
r = 12
r = 18
ASPP
H/4
H/8
Local Planar Guidance (lpg8x8)
upconv
H/4
conv
H/4
Local Planar Guidance (lpg4x4)
upconv
H/2
H/2
conv
H/2
Local Planar Guidance (lpg2x2)
upconv
H
1x1 Reduction (reduc1x1)
H
ds/4
ds/2
H
H
H
H
conv
H
$\tilde{c}^{8\times8}$
$\tilde{c}^{4\times4}$
$\tilde{c}^{2\times2}$
$\tilde{c}^{1\times1}$
$\tilde{d}$
Skip-connection
DFE
Dense Feature Extractor
conv
3x3 conv, s1
1x1 conv, s1
upconv
2x upsample
3x3 conv, s1
ds
Down Sampling
Concat


H/k
Local Planar Guidance
H
H/k
1x1 reduction
1st , 2nd channels
3rd channel
Equation 2
Sigmoid
× κ
n_1, n_2, n_3
n_4
H/k
Equation 1
H

Capítulo 28

“Deep Ordinal Regression Network for Monocular Depth Estimation”

El artículo propone una red de regresión ordinal profunda (DORN) para estimar la profundidad a partir de una sola imagen, abordando el problema mal planteado de la estimación de profundidad monocular (MDE). En lugar de tratar MDE como un problema de regresión estándar con pérdida de error cuadrático medio (MSE), que converge lentamente y produce soluciones subóptimas, DORN discretiza los valores de profundidad usando una estrategia de discretización de espaciado creciente (SID) en espacio logarítmico. Esto transforma MDE en un problema de regresión ordinal, optimizado con una pérdida ordinal que considera la correlación ordenada de los valores discretos, logrando mayor precisión y convergencia más rápida.

La arquitectura de la red evita el submuestreo excesivo mediante convoluciones dilatadas, manteniendo mapas de características de alta resolución. Incluye un módulo de comprensión de escenas con atrous spatial pyramid pooling (ASPP) para capturar información multiescalar y un codificador de imagen completa eficiente que reduce significativamente los parámetros en comparación con enfoques tradicionales. La red se entrena de extremo a extremo sin refinamiento por etapas.

Ha sido evaluado en los conjuntos de datos KITTI, Make3D y NYU Depth v2. Los estudios de ablación confirman la importancia de SID, la pérdida ordinal y el codificador de imagen completa. Aunque robusto a diferentes números de intervalos de discretización, el rendimiento óptimo se logra con 40 a 120 intervalos. DORN ofrece una solución eficiente y precisa para MDE, con potencial para extenderse a otros problemas de predicción densa.

Fórmula:

Es la función de pérdida ordinal utilizada para entrenar la red, ya que es el núcleo de la reformulación del problema de estimación de profundidad como un problema de regresión ordinal.

$$\mathcal{L}(x,\theta) = -\frac{1}{N}\sum_{w=0}^{W-1}\sum_{h=0}^{H-1}\psi(w,h,x,\theta)$$

$$\psi(h,w,x,\theta) = \sum_{k=0}^{l_{(w,h)}-1}\log(\mathcal{P}^{k}_{(w,h)}) + \sum_{k=l_{(w,h)}}^{K-1}(1-\log(\mathcal{P}^{k}_{(w,h)}))$$

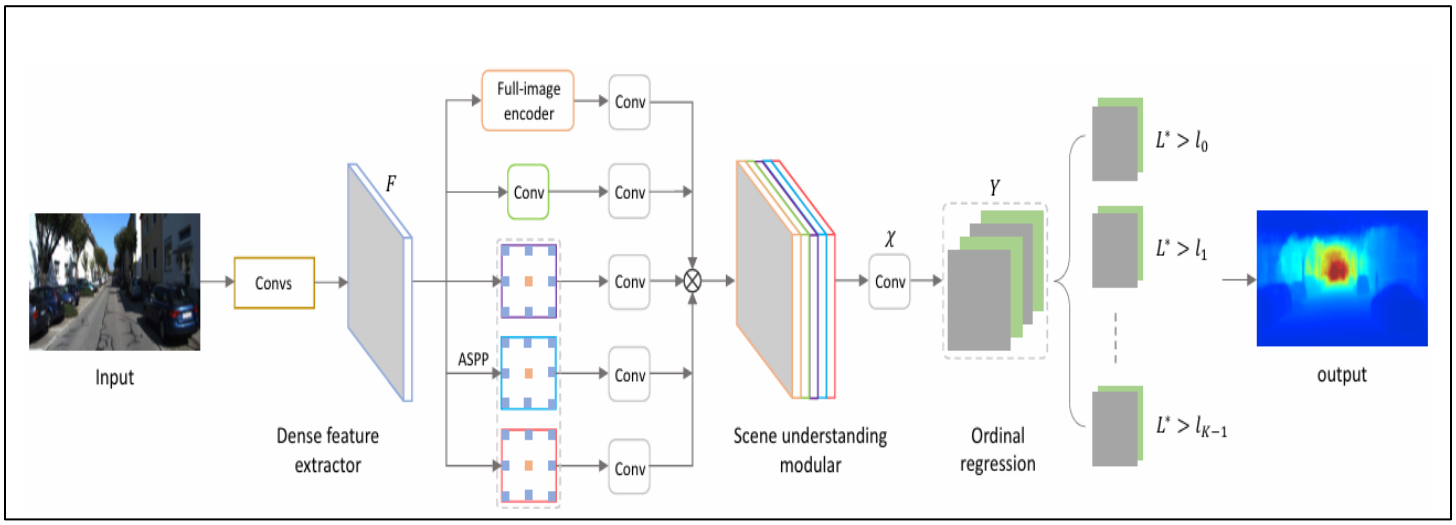
$$\mathcal{P}^{k}_{(w,h)} = P(\hat{l}_{(w,h)} > k|x,\theta)$$

Otra de las fórmulas en la estrategia de discretización es:

$$UD\colon t_i = \alpha + (\beta-\alpha) * i|K$$

$$SID\colon t_i = e^{\log(\alpha)+\frac{\log(\beta|\alpha)*i}{K}}$$

Esquema:


Input
Convs
F
Dense feature extractor
Full-image encoder
Conv
Conv
Conv
Conv
Conv
ASPP
Conv
Conv
Scene understanding modular
χ
Conv
Y
Ordinal regression
$L^* > l_0$
$L^* > l_1$
$L^* > l_{K-1}$
output

Capítulo 29

“Is Pseudo-Lidar needed for Monocular 3D Object detection?”

El artículo presenta DD3D, un detector monocular 3D de una sola etapa que combina la escalabilidad de métodos pseudo-lidar con la simplicidad y generalización de detectores de extremo a extremo. A diferencia de los métodos pseudo-lidar, que convierten imágenes en nubes de puntos 3D mediante estimación de profundidad intermedia, **DD3D** predice directamente cajas delimitadoras 3D y mapas de profundidad densos, evitando la dependencia de redes de profundidad separadas y su sobreajuste a errores de profundidad. Su arquitectura, basada en **FCOS** con una red de pirámide de características (FPN), incluye cabezales para clasificación, detección 2D y 3D, compartiendo parámetros para maximizar la transferencia desde el pre-entrenamiento en profundidad.

DD3D se pre-entrena en el conjunto de datos DDAD15M (15M de imágenes) para estimación de profundidad densa, utilizando datos LiDAR proyectados sin etiquetas humanas, y se ajusta finamente para detección 3D en KITTI-3D y nuScenes. Comparado con pseudo-lidar, DD3D es más simple, no requiere ajuste fino de profundidad en dominio y generaliza mejor, mostrando menor pérdida de precisión entre validación y prueba.

El diseño consciente de la cámara, que ajusta predicciones según parámetros intrínsecos, y la pérdida disentangled L1 estabilizan el entrenamiento. Los experimentos confirman que el pre-entrenamiento en profundidad supera al pre-entrenamiento en detección 2D (COCO), y la escalabilidad con datos no etiquetados mejora el rendimiento, haciendo de DD3D una solución práctica y robusta para detección 3D monocular en aplicaciones como conducción autónoma.

Fórmula:

Tenemos la función de pérdida total utilizada para entrenar el modelo DD3D, ya que encapsula los componentes clave para optimizar la detección de objetos 3D monoculares y la predicción de profundidad densa.

$$\mathcal{L}_{DD} = \mathcal{L}_{2D} + \mathcal{L}_{3D} + \mathcal{L}_{conf}$$

Perdida 2D se adopta del modelo FCOS [59] y se compone de tres términos:

$$\mathcal{L}_{2D} = \mathcal{L}_{reg} + \mathcal{L}_{cls} + \mathcal{L}_{ctr}$$

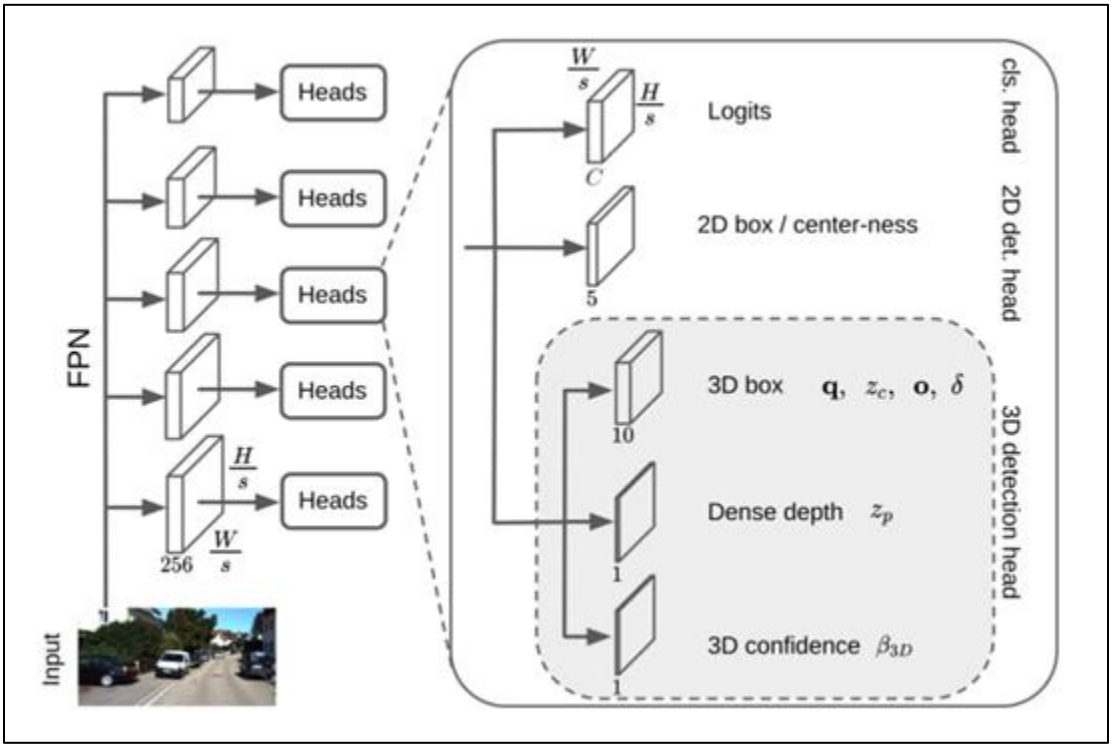
Perdida 3D utiliza una pérdida L1 disentangled [57] para la regresión de cajas 3D, definida como:

$$\mathcal{L}_{3D}(B^*, \hat{B}) = \frac{1}{8}||B^* - \hat{B}||_1$$

Pérdida de confianza, es una pérdida de entropía cruzada binaria que relaciona la confianza predicha (p_{3D}) con un objetivo surrogate basado en el error de la predicción 3D:

$$p^*_{3D} = e^{-\frac{1}{T}\mathcal{L}_{3D}(B^*, \hat{B})}$$

Esquema:


Heads
Heads
Heads
Heads
Heads
FPN
Input
256
Logits
2D box / center-ness
3D box
Dense depth
3D confidence
cls. head
2D det. head
3D detection head

Capítulo 30

“Deep Line Encoding for Monocular 3D Object Detection and Depth Prediction”

El artículo introduce Deep Line Encoding, un método que mejora la percepción de profundidad en imágenes RGB únicas para detección de objetos 3D y predicción de profundidad en escenarios de conducción autónoma. Propone explotar líneas rectas y puntos de fuga como pistas geométricas clave, ya que indican la pendiente del terreno y el diseño 3D de la escena. Para ello, utiliza el Deep Hough Transform en mapas de características de redes profundas, transformando líneas en un espacio paramétrico donde la agregación de características codifica la semántica de la línea y la posición de votación indica parámetros algebraicos (ángulo y distancia).

Se introduce un módulo novedoso de Line Pooling que selecciona y comprime las líneas más relevantes en un vector compacto, mejorando la eficiencia al descartar agregaciones irrelevantes. Este vector se fusiona con mapas de coordenadas y características del backbone, integrándose en frameworks existentes como VisualDet3D y GAC para detección 3D y predicción de profundidad, respectivamente, sin supervisión adicional.

Evaluado en el conjunto de datos KITTI, Deep Line Encoding mejora el estado del arte en detección 3D monocular, un aumento de 2.58 puntos sobre VisualDet3D, y en predicción de profundidad, logra un sqErrorRel de 2.22, superando a GAC. Los estudios de ablación confirman la importancia del mapa de coordenadas y la operación softmax en Line Pooling. La visualización muestra que las líneas seleccionadas se alinean con estructuras relevantes como barandillas y puntos de fuga. El método es genérico y no depende de modelos externos de profundidad.

Fórmula:

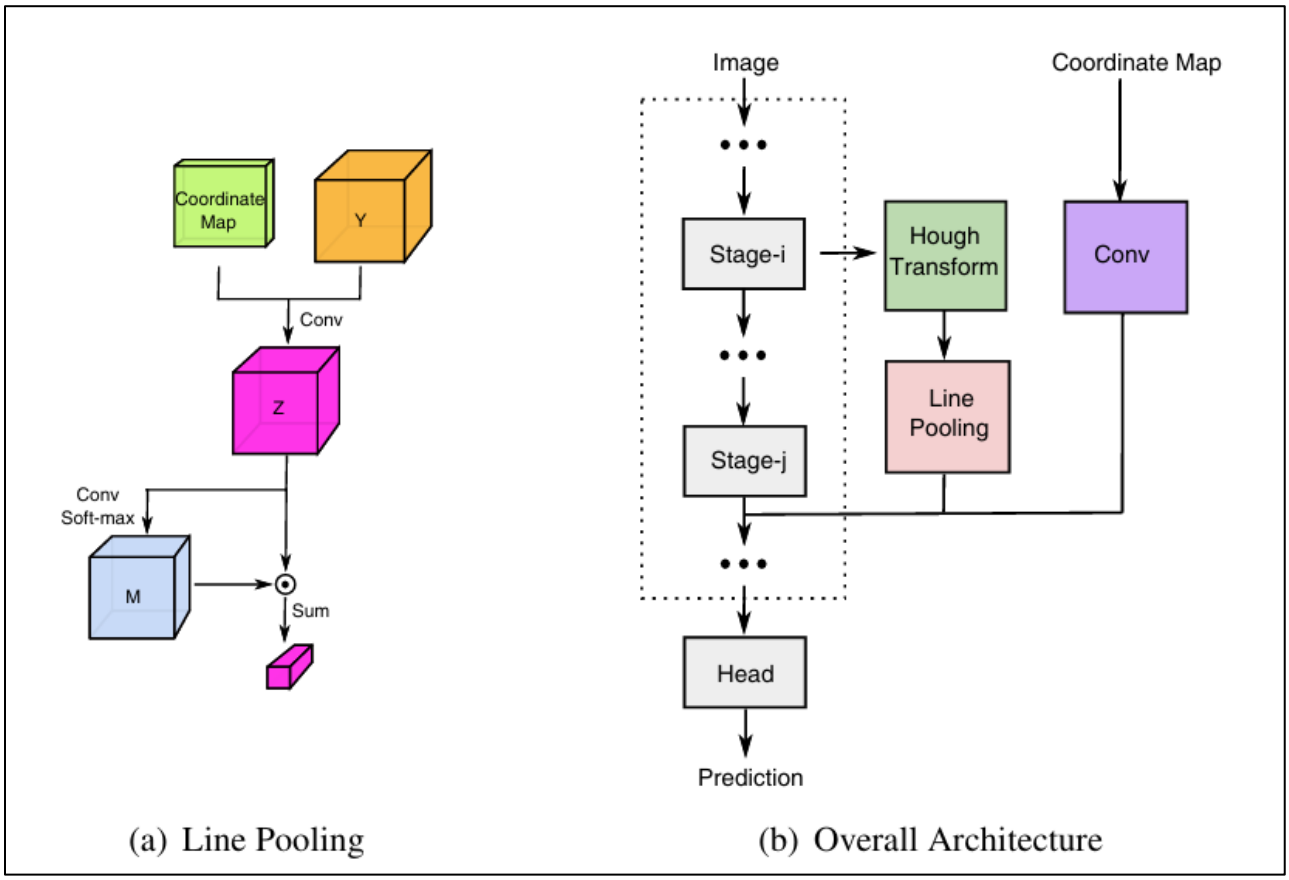
Se tiene la ecuación que describe la transformada de Hough profunda utilizada en el módulo de codificación de líneas profundas (Deep Line Encoding), ya que es el núcleo del método propuesto para explotar la información de líneas rectas en escenas para mejorar la percepción de profundidad.

$$Y(\theta, \rho) = \sum_{(x,y)\epsilon^{l}} X(x, y)$$

Como fórmula complementaria tenemos la estimación de profundidad simplificada definida como:

$$Z = \frac{fY}{y}$$

Esquema:


Coordinate Map
Y
Conv
Z
Conv Soft-max
M
Sum
Image
Coordinate Map
Stage-i
Hough Transform
Conv
Line Pooling
Stage-j
Head
Prediction
(a) Line Pooling
(b) Overall Architecture