CEIG – Spanish Computer Graphics Conference (2021)
A. Chica and L. Ortega (Editors)

Generation process of intrinsic images dataset through physically-based rendering

Ignacio M. Rodríguez, Alfonso López [1], J. Roberto Jimenez-Perez [1], Francisco R. Feito [1], Lidia Ortega [1] y Juan M. Jurado [1]

[1] Computer Graphics and Geomatics Group, University of Jaén, Spain

Abstract

El problema denominado Intrinsic Image Decomposition sigue siendo un desafío por resolver en informática gráfica. Aunque el uso de arquitecturas de aprendizaje profundo supondría un avance significativo, los conjuntos de datos de entrenamiento utilizados son aún reducidos. En este estudio se presenta una metodología para la generación de imágenes y su descomposición en varios canales haciendo uso del motor de renderizado Mitsuba2. Para ello, se ha modelado un escenario natural en el que coexisten distintos tipos de vegetación sobre un terreno. En torno a este escenario, se define una trayectoria sobre la que orbita la cámara para generar un conjunto de imágenes desde distintos puntos de vista de forma automática. Como resultado, se proporcionan conjuntos de datos obtenidos a partir de entornos naturales sintéticos formados por las siguientes capas para cada imagen: mapa de normales, iluminación, albedo y mapa de profundidad. Este desarrollo supone un punto de partida para el estudio del cálculo de la iluminación en entornos reales complejos mediante enfoques basados en aprendizaje profundo.

CCS Concepts

• ***Computing methodologies*** → *Computer graphics; Intrinsic decomposition; Rendering;*

1. Introducción

Intrinsic Descomposition es un proceso que descompone una imagen en un conjunto de éstas, donde cada una representa una propiedad diferente de la escena. El modelo clásico propuesto por Barrow y Tenembaum en [BTHR78], especificaba que cada píxel de la imagen original sería el resultado de la multiplicación de dos componentes: *shading* y reflectancia (o albedo), dando como lugar:

$$I(x,y) = Ref(x,y) \cdot Sha(x,y) \tag{1}$$

El problema radica en separar una imagen original en la estimación de su *shading* y reflectancia (o albedo), donde el *shading* se representa normalmente en tonos de grises, capturando la interacción de la iluminación con la escena, y el albedo en color, capturando las propiedades cromáticas de la superficie del objeto. Nuestra intención con la selección de este modelo, es la de comenzar con un sistema básico que posteriormente se va enriqueciendo. Por esta razón, usamos una versión simplificada de la *ecuación de rendering* [SDSY17], descrita como la integral del producto del BSDF por el campo de radiancia incidente.

A pesar de las grandes ventajas que supondría resolver esta descomposición para el campo de la Visión por Computador, todavía sigue siendo un reto por resolver. Actualmente, las propuestas más prometedoras se basan en el uso de redes neuronales, pero existen numerosos problemas en los *datasets* de entrenamiento que no permiten tener una buena función de evaluación para estas redes.

En este estudio proponemos un proceso de generación de imágenes realistas, cuyos materiales aplicados en los modelos sean adquiridos del mundo real. Como consecuencia, dichos retratos podrán ser usados como entrenamiento de futuras arquitecturas de aprendizaje profundo especializadas en el Intrinsic Image Descomposition, debido a que el motor de renderizado nos permitirá extraer tanto el albedo como el shading de la escena. Para ello se usará Mitsuba2 [NDVZJ19], un reciente motor de renderizado orientado a la investigación escrito en C++ 17, el cual nos permite renderizar una escena con iluminación global, definir los colores mediante su espectro electromagnético o trabajar con la polarización de los rayos, logrando así generar imágenes fotorrealistas.

2. Estado del arte

Aunque el Intrinsic Descomposition ha sido reconocido como un importante problema dentro de la Visión por Computador, ha experimentado un progreso limitado debido a su dificultad intrínseca. Los primeros trabajos en este campo estaban enfocados en la introducción de restricciones en la imagen y en el uso de métodos de optimización. Por ejemplo, Tappen et al. en [TFA05] o [MLKS04a] utilizaban la información de color y clasificadores binarios para distinguir las derivadas causadas por el albedo de las causadas por el shading o Shen et al. que proponían identificar píxeles con la misma reflectancia a partir de sus vecinos [STL08].

DOI: 10.2312/ceig.20211362

Posteriores investigaciones añadieron más restricciones físicas a este problema inverso. En algunos trabajos, la información de profundidad también se añade, llegando a lograr resultados prometedores. Otros trabajos se enfocaron en la iluminación variando secuencia de imágenes a partir de cámaras de vídeo estáticas [Wei01] [MLKS04b] [LB15], donde la reflectancia es constante en las distintas imágenes, reduciendo la complejidad del problema.

A diferencia del trabajo clásico anterior, la investigaciones más recientes se basan en el uso de arquitecturas de aprendizaje profundo. El gran éxito de las Redes Neuronales Convolucionales (CNN) en la Visión Computacional hizo que la comunidad se interesara en resolver el Intrinsic Descomposition a través de estas arquitecturas. Los primeros enfoques son introducidos por Narihira et al. [NMY15].

A pesar del éxito de estas arquitecturas, existe una opinión común de que los datasets usados son los responsables del bajo rendimiento que estas proporcionan. El principal problema respecto a los *datasets* para la estimación de imágenes intrínsecas oscila entre el realismo de las propiedades de iluminación física de la escena (IIW) [BBS14] y la cantidad de imágenes proporcionadas (MIT) [GJAF09], las cuales deben ser suficientes para el entrenamiento de arquitecturas de aprendizaje profundas.

Debido a que estas arquitecturas tienen la necesidad de realizar un periodo de entrenamiento, deben utilizar conjuntos de muestras válidas, necesarias para obtener un mejor rendimiento en el futuro. Algunas de las muestras que podemos destacar son las siguientes:

- *MIT Intrinsic* [GJAF09]: Primer *dataset* en este dominio, pero no es sintético, es decir, fue generado a partir del mundo real en entornos con condiciones muy controladas. Tiene 20 objetos con 11 condiciones de luz diferentes (220 muestras).
- *MPI Sintel* [BWSB12]: *Dataset* sintético basado en una película de animación. Este contiene 18 escenas con 50 fotogramas cada una, excepto una de 40 (890 muestras). El problema de este dataset es que presenta un coloreado poco natural, donde los tonos tienen sesgos de color principalmente en azul y marrón.
- *IIW: Intrinsic Images in the Wild* [BBS14]: *Dataset* de 5230 imágenes que proporciona juicios entre pares de reflectancia, los cuales presentan un mapa espacial coherente de reflectancia. El problema de entrenar redes a partir de estas muestras es que las redes presentan una estimación demasiado suave con falta de variación de textura.

El problema estos y otros datasets es que suelen presentar incoherencias en la luz, los fondos de los escenarios no están muy diversificados, o el conjunto de imágenes generadas suele ser pequeño. Es por ello que nuestro objetivo es dotar a la comunidad una nueva manera de crear escenarios, en los cuales cumpliremos los anteriores requisitos mencionados.

3. Creación del escenario

La mayoría de los datasets trabajan con imágenes en las que se suele mostrar un único objeto central, incrustadas en un escenario básico. Tanto con los objetos como con los fondos suele existir un conjunto de éstos, los cuales varían entre sí. Normalmente, el escenario suele ser una caja de Cornell o un fondo sencillo. Esta simplificación de la escena a veces puede llegar a ser perjudicial para el entrenamiento de la red. Por ello proponemos una generación del escenario, que imita elementos de la vida real, generando un terreno y colocando objetos con alto nivel de detalle.

Para la generación del terreno proponemos el uso de escenarios en la naturaleza, consiguiendo un fondo robusto con el que interactuar a diferencia de otros datasets. A partir de mapas de altura podemos modelar la superficie de un terreno del mundo real en nuestra escena. Posteriormente, se añaden los modelos hasta lograr el detalle deseado. Todo esto se realiza desde Blender, con un plugin que exporta la escena de Blender al formato de archivo de Mitsuba2. Una de las principales limitaciones es el tipo de archivo que lee, wavefront o PLY, por lo que, no podemos usar muchas de las funcionalidades potentes de modelado que ofrece Blender. Mitsuba2 sólo es capaz de importar los materiales más básicos, como el *diffuse BSDF*, *image texture* o *glass BSDF*, aunque se está trabajando en permitir más, como el *principle BSDF*. No obstante, los materiales importados en Mitsuba2 pueden posteriormente ser ajustados. Hemos de recordar que una de las funcionalidades más relevantes que presenta este motor, es que permite trabajar con muchas categorías de BSDFs, uno de los más importantes es el llamado *measured* el cual permite cargar BSDF adquiridas del mundo real, con mediciones obtenidas a partir de un goniofotómetro, que especifican de manera detallada y parametrizada cómo interacciona la luz con un material. Posteriormente, estas mediciones pueden ser procesadas, tal y como indican Dupuy et al., [DJ18]. Además, estos ficheros de materiales del mundo real, se pueden obtener de numerosas bases de datos.

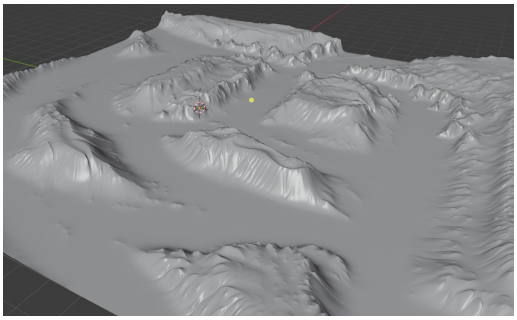
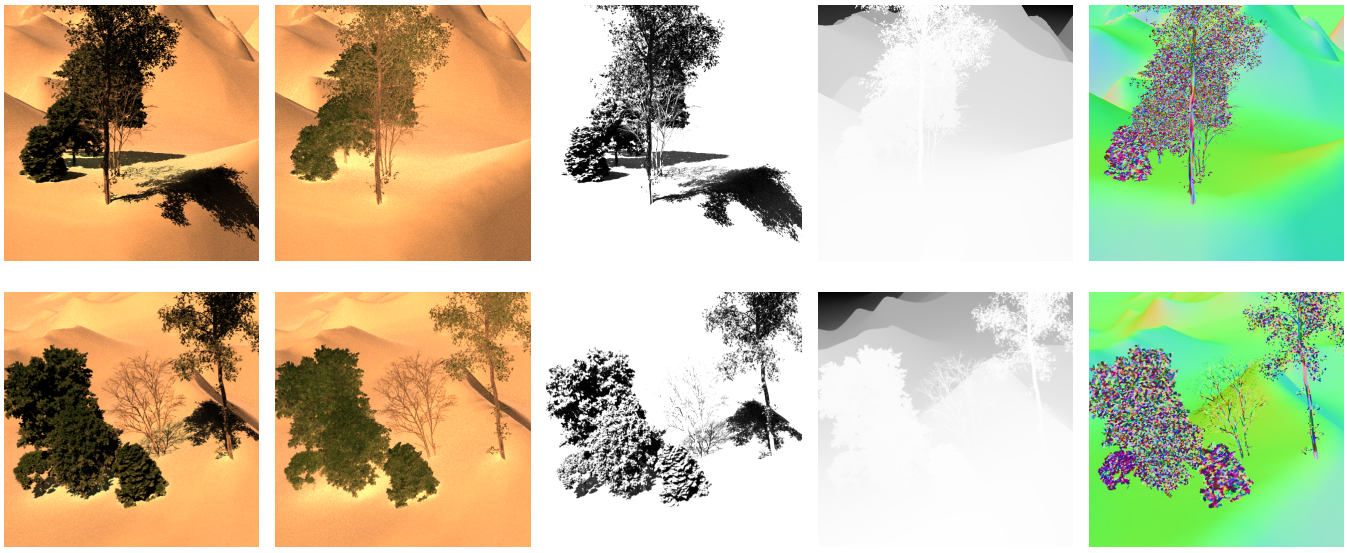


Figure 1: *Terreno de ejemplo creado a partir de un Heightmap.*

Una vez el fichero XML ha sido generado, y hemos especificado los BSDFs que se usarán para cada tipo de objeto, tendríamos todo lo necesario para el renderizado de la escena, obteniendo muestras de entrenamiento para futuras redes de aprendizaje profundo.

Para la generación de dichas muestras, necesitaremos al menos de los siguientes tipos de imágenes: La imagen original, la imagen de reflectancia (o albedo) y la imagen de sombreado (o shading). Para la obtención de estas imágenes necesitaremos especificarle a Mitsuba2 como debe renderizar la escena, para ello, necesitaremos implementar la forma en la que vamos a tratar la luz para renderizar la escena, agregando nuevos plugins.

Los *Integrators* usados en este trabajo son variaciones desarrolladas por los autores, a partir de los *Integrators* **volpath** y **volpathmis** que brinda Mitsuba, permitiendo la obtención tanto del



(a) *imagen resultado* **(b)** *material* **(c)** *iluminación* **(d)** *profundidad* **(e)** *normales*

Figure 2: *Ejemplo de una muestra obtenida a partir del motor de renderizado Mitsuba2.*

sombreado como de la reflectancia respectivamente. Estas variaciones normalmente tratan de la eliminación de la componente de emisión o de reflectancia. El no uso de los Integrators **path** y **direct** es debido a que estos no pueden trabajar con objetos translúcidos.

Otro aspecto a tener en cuenta es el espacio de color que usaremos en cada tipo de muestra. Por ejemplo, el modelo clásico del Intrinsic Descomposition [BTHR78] dice que el albedo se representa en color, mientras que el shading se hace en tonos de blanco y negro. Esto lo debemos de tener en cuenta a la hora de obtener las muestras, ya que Mitsuba2 nos permite obtener los resultados en cualquier espacio de color, desde monocromático, hasta RGB o su espectro electromagnético.

También se necesitará especificar la iluminación de la escena. Debido a que crearemos escenas de naturaleza, una con iluminación direccional bastará, aunque se podría usar otros modelos más complejos si se quiere tener en cuenta la dispersión de radiancia de la atmósfera. Debemos tener en cuenta que la intensidad de esta solo afectará al shading, debido a que el albedo representa el color intrínseco del objeto, y la varianza de la iluminación no determina cómo debe de afectar al albedo.

Para la obtención de la imagen original solo bastaría con el renderizado de la escena con cualquiera de otros tipos de Integrators de los que dispone Mitsuba2, o también podría ser obtenida posteriormente, como resultado de la multiplicación de la imagen albedo y shading.

Además de obtener el albedo y shading, algunas redes también hacen uso de imágenes más especiales como la profundidad o mapa de normales. Esto también es proporcionado por Mitsuba2, por lo que se pueden generar en caso de ser necesario.

En nuestro script de generación de muestras, a la hora de generar la imágenes, se especifica un centro de interés (COI) de la cámara virtual y la altura (h) y distancia (d) que esta tendrá de dicho punto. Según el número de divisiones indicadas, obtendremos más o menos muestras respecto al COI especificado. De dichas muestras se obtendrán las imágenes que hayan sido detalladas (albedo, shading, profundidad, normales, etc.), proporcionando todo lo necesario para el entrenamiento de futuras redes. Cabe mencionar que Mitsuba2 también tiene integración en Python, por lo que la generación del script se ha realizado desde este lenguaje de programación.

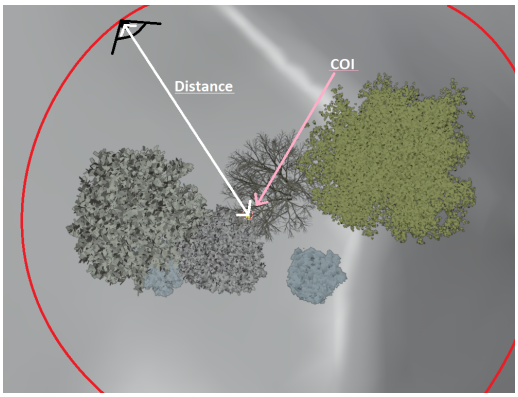


Figure 3: *Movimiento a realizar por la cámara virtual para la obtención de múltiples muestras a partir de una escena.*

4. Resultados y Discusión

En esta sección se muestran los resultados generados tras aplicar la metodología propuesta. La generación de imágenes que componen nuestro conjunto de datos tiene en cuenta un renderizado fotorrealista al simular propiedades físicas del mundo real, tratando fenómenos como la reflectancia de los materiales y la polarización de los rayos de luz. La Figura 2 muestra un ejemplo del conjunto

de imágenes generadas para un punto de vista dado en el escenario sintético. Se han obtenido resultados realistas, utilizando modelos geométricos con bajo nivel de detalle. En este caso, además de obtener la información del albedo 2b y sombreado 2c, también hemos obtenido información de profundidad 2d y de normales 2e.

La generación del conjunto de imágenes se ha obtenido mediante la rotación de una cámara alrededor del escenario creado, tomando 50 muestras alrededor de este, y cada muestra a su vez está compuesta por cinco imágenes (real, albedo, reflectancia, profundidad y normal). Como resultado el conjunto de datos provisto lo componen un total de 250 imágenes de 512x512 píxeles.

Para incluir la profundidad se ha tenido que hacer una transformación adicional al resultado proporcionado por Mitsuba. Este framework proporciona información de distancias reales, que hemos normalizado para poder convertirlas a un formato de imagen. Las zonas más claras corresponden con distancias más cercanas a la cámara. Esta transformación hay que tenerla en cuenta a la hora de trasladar el conjunto de imágenes a la red neuronal para homogeneizar el sistema de distancias y evitar así perturbar su aprendizaje.

En las imágenes resultantes comprobamos que las hojas del árbol utilizado tienen una gran variedad de normales. Esta característica es común en entornos naturales y, por tanto, será un tema de estudio interesante comprobar el comportamiento de las redes neuronales a la hora de incorporar el conocimiento procedente de su geometría.

5. Conclusión y Trabajo Futuro

Este estudio ha proporcionado un método para obtener conjuntos de imágenes para el entrenamiento de redes neuronales orientadas a resolver el problema de Intrinsic Image Descomposition. El motor de renderizado Mitsuba2 nos ha posibilitado la generación de imágenes altamente realistas, así como el tratamiento de materiales BRDF extraídos del mundo real. La descomposición de la escena se ha llevado a cabo haciendo uso de algunas de las utilidades proporcionadas por el motor de renderizado. Para la generación del dataset se provee de un script con el que obtener múltiples capturas en torno a punto de interés definido en la escena sintética.

Futuras investigaciones se centrarán en la creación de nuevos datasets a partir de este motor de renderizado, donde las escenas tengan diferentes niveles de detalle, y los materiales aplicados se obtengan a partir del mundo real (BSDF). También se debería de tener en cuenta una futura aplicación de iluminación realista en vez de una direccional, como por ejemplo la propuesta por Ron y Alan et al. [DWA04] que enriquecería considerablemente las muestras. Además de lo anterior, una vez se tenga un conjunto de muestras lo suficientemente amplio y diverso, también se podrían realizar pruebas pertinentes de dichas muestras con redes de aprendizaje profundo.

Agradecimientos

Este trabajo ha sido parcialmente apoyado por el Ministerio de Ciencia, Innovación y Universidades a través del proyecto de investigación TIN2017-84968-R.

References

[BBS14] BELL S., BALA K., SNAVELY N.: Intrinsic images in the wild. *ACM Trans. Graph. 33*, 4 (July 2014). doi:10.1145/2601097.2601206. 2

[BTHR78] BARROW H., TENENBAUM J., HANSON A., RISEMAN E.: Recovering intrinsic scene characteristics. *Comput. Vis. Syst 2*, 3-26 (1978), 2. 1, 3

[BWSB12] BUTLER D. J., WULFF J., STANLEY G. B., BLACK M. J.: A Naturalistic Open Source Movie for Optical Flow Evaluation. In *Computer Vision – ECCV 2012* (Berlin, Heidelberg, 2012), Fitzgibbon A., Lazebnik S., Perona P., Sato Y., Schmid C., (Eds.), Lecture Notes in Computer Science, Springer, pp. 611–625. doi:10.1007/978-3-642-33783-3_44. 2

[DJ18] DUPUY J., JAKOB W.: An adaptive parameterization for efficient material acquisition and rendering. *ACM Trans. Graph. 37*, 6 (Dec. 2018). doi:10.1145/3272127.3275059. 2

[DWA04] DROR R. O., WILLSKY A. S., ADELSON E. H.: Statistical characterization of real-world illumination. *Journal of Vision 4*, 9 (09 2004), 11–11. doi:10.1167/4.9.11. 4

[GJAF09] GROSSE R., JOHNSON M. K., ADELSON E. H., FREEMAN W. T.: Ground truth dataset and baseline evaluations for intrinsic image algorithms. In *2009 IEEE 12th International Conference on Computer Vision* (Sept. 2009), pp. 2335–2342. ISSN: 2380-7504. doi:10.1109/ICCV.2009.5459428. 2

[LB15] LAFFONT P.-Y., BAZIN J.-C.: Intrinsic decomposition of image sequences from local temporal variations. In *2015 IEEE International Conference on Computer Vision (ICCV)* (2015), pp. 433–441. doi:10.1109/ICCV.2015.57. 2

[MLKS04a] MATSUSHITA Y., LIN S., KANG S. B., SHUM H.-Y.: Estimating intrinsic images from image sequences with biased illumination. In *Computer Vision - ECCV 2004* (Berlin, Heidelberg, 2004), Pajdla T., Matas J., (Eds.), Springer Berlin Heidelberg, pp. 274–286. doi:10.1109/CVPR.2006.114. 1

[MLKS04b] MATSUSHITA Y., LIN S., KANG S. B., SHUM H.-Y.: Estimating intrinsic images from image sequences with biased illumination. In *Computer Vision - ECCV 2004* (Berlin, Heidelberg, 2004), Pajdla T., Matas J., (Eds.), Springer Berlin Heidelberg, pp. 274–286. 2

[NDVZJ19] NIMIER-DAVID M., VICINI D., ZELTNER T., JAKOB W.: Mitsuba 2: A retargetable forward and inverse renderer. *ACM Trans. Graph. 38*, 6 (Nov. 2019). doi:10.1145/3355089.3356498. 1

[NMY15] NARIHIRA T., MAIRE M., YU S. X.: Direct Intrinsics: Learning Albedo-Shading Decomposition by Convolutional Regression. In *Proceedings of the IEEE international conference on computer vision* (2015), pp. 2992–2992. doi:10.1109/ICCV.2015.342. 2

[SDSY17] SHI J., DONG Y., SU H., YU S. X.: Learning non-lambertian object intrinsics across shapenet categories. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (July 2017). 1

[STL08] SHEN L., TAN P., LIN S.: Intrinsic image decomposition with non-local texture cues. In *2008 IEEE Conference on Computer Vision and Pattern Recognition* (June 2008), pp. 1–7. ISSN: 1063-6919. doi:10.1109/CVPR.2008.4587660. 1

[TFA05] TAPPEN M., FREEMAN W., ADELSON E.: Recovering intrinsic images from a single image. *IEEE Transactions on Pattern Analysis and Machine Intelligence 27*, 9 (Sept. 2005), 1459–1472. Conference Name: IEEE Transactions on Pattern Analysis and Machine Intelligence. doi:10.1109/TPAMI.2005.185. 1

[Wei01] WEISS Y.: Deriving intrinsic images from image sequences. In *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001* (July 2001), vol. 2, pp. 68–75 vol.2. doi:10.1109/ICCV.2001.937606. 2