

Materiales “SSS Fast (mi)” y “SSS Fast Skin (mi)” (3ds Max)

(traducido por [Gabriel Gazzán](#) para [A+ escuela de artes visuales](#))

Nota: los nombres que aparecen en **negrita** son los nombres originales de Mental Ray, el texto que aparece **resaltado** a su lado corresponde al nombre tal como aparece en la interfaz de 3ds Max.

Fenómenos

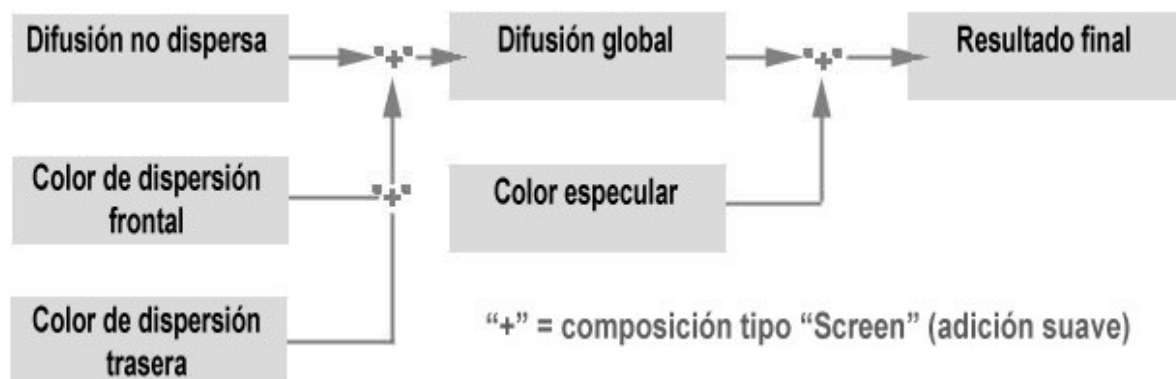
Los Fenómenos suministrados se documentan primero porque son suficientes para la mayoría de los usos prácticos. Los usuarios expertos podrán crear sus propios Fenómenos utilizando estos como referencia o usar los componentes de shaders tal como se describen en la página `misss:advanced`. Estos Fenómenos utilizan estructuras para agrupar los parámetros de entrada en grupos lógicos, los que aparecerán automáticamente juntos en las interfaces de las distintas integraciones OEM de Mental Ray.

misss_fast_simple_phen **SSS Fast Material (mi)**

Este Fenómeno automáticamente (a menos que se explicito lo contrario) genera los mapas de iluminación requeridos, asigna la información de iluminación y sombrea los objetos como objetos translúcidos genéricos con dispersión bajo su superficie. La dispersión es dividida en dos partes, una para el lado frontal del objeto y otra para el trasero.

Se podría decir que el sombreador trabaja componiendo una sobre otra, en capas, las varias contribuciones de luz para obtener el resultado final. Es importante tener en mente las siguientes reglas básicas:

- El mapa de relieve (bump) sólo afecta a las capas *specular* y *diffuse_color*. La dispersión bajo la superficie sucede por debajo y es por tanto intrascendente que la superficie tenga rugosidades o no.
- El *overall_color* afecta a todas las contribuciones de difusión, pero no a la especularidad:



Flujo del color


```

material "misss_fast_simple_phen" (
    color texture "lightmap",
    color texture "depthmap",
    string        "lightmap_group",
    scalar        "lightmap_size",
    integer       "samples",
    shader        "bump",
    struct "d" {
        color      "ambient",
        color      "overall_color",
        color      "diffuse_color",
        scalar     "diffuse_weight",
        color      "front_sss_color",
        scalar     "front_sss_weight",
        scalar     "front_sss_radius",
        color      "back_sss_color",
        scalar     "back_sss_weight",
        scalar     "back_sss_radius",
        scalar     "back_sss_depth"
    },
    struct "s" {
        color      "specular",
        scalar     "exponent"
    },
    struct "a" {
        scalar     "lightmap_gamma",
        boolean    "indirect",
        scalar     "scale_conversion",
        scalar     "scatter_bias",
        scalar     "falloff",
        boolean    "screen_composit"
    },
    integer       "mode",          # light selection mode 0..2
    array light   "lights"
)

```

lightmap y **depthmap** no son usualmente especificados. Estos son parámetros especiales usados por los shaders para transmitir el nombre de los mapas de iluminación creados automáticamente a otros shaders. Los usuarios expertos pueden asignar texturas escribibles y dejar los parámetros *lightmap_group* y *lightmap_size* sin definir. Ver la página *misss:autolightmap*.

lightmap_group **Scatter group** es el nombre del grupo de dispersión. Internamente se creará un conjunto de mapas de iluminación y profundidad a los que se asignará este nombre. Todos los objetos cuyas dispersiones de luz deben interactuar mutuamente deben pertenecer al mismo grupo de dispersión. Para ahorrar memoria, usar la menor cantidad posible de grupos de dispersión. Las manos, rostro, etc. de una persona pueden usar el mismo grupo de dispersión; inclusive los rostros y manos de otros personajes pueden pertenecer al mismo grupo de dispersión. En general, se deben utilizar distintos grupos de dispersión sólo en caso de que utilizar el mismo grupo pueda causar problemas visibles debido a que los objetos dispersen incorrectamente su luz entre ellos. Dos personajes dándose un apretón de manos, por ejemplo, necesitarían tener sus manos en distintos grupos de dispersión. (Ver también la nota técnica sobre render en red en la página *misss:network*.)

lightmap_size **Lightmap size** es el tamaño del mapa de iluminación expresado como porcentaje del tamaño del render. El mapa de iluminación es creado en el momento del render final en función de los pixels de la imagen. 50% o menos es usualmente suficiente (y es el valor predefinido), pero si aparecieran defectos en los bordes, puede ser necesario aumentar ese valor.

samples **Number of samples** establece cual es el máximo de muestras del mapa de iluminación que se toman en cuenta por cada rayo renderizado, idealmente debería ser una potencia de 2. Probablemente 32 es el mínimo valor útil, 128 se considera ya un valor alto. Sin embargo, si los radios de dispersión son grandes podría ser necesario aumentar el número de muestras para evitar la aparición de ruido en la imagen.

bump **Bump shader** permite asignar un shader que perturbe el valor de la normal para obtener rugosidad. El componente de color aportado por el shader (si existiera) no será utilizado.

ambient **Ambient / Extra light** es el componente ambiental o cualquier luz difusa adicional aportada por cualquier shader. Esta luz también será tomada en cuenta en el mapa de iluminación y dispersada como cualquier otra fuente lumínica y es, por tanto, el lugar perfecto para agregar cualquier iluminación HDRI. La luz HDRI puede basarse en mapas de entorno u oclusión.

overall_color **Overall diffuse coloration** es el color global de difusión. Todos los colores de difusión aparecen debajo se multiplican por este color global para obtener el resultado final. Este parámetro es útil para teñir el color de difusión resultante del cálculo en un sentido deseado o para aplicar mapas que atenúen toda la iluminación difusa. Lunares en la piel, por ejemplo, bloquean la gran mayoría de la luz subyacente (aún la dispersa) y tendría sentido incluirlos aquí, al igual que suciedad, cejas pintadas y cosas por el estilo.

diffuse_color **Unscattered diffuse color** es el color y **diffuse_weight** **Unscattered diffuse weight** la influencia (un simple multiplicador para facilitar la manipulación) de la iluminación difusa normal, de tipo Lambert, que se compone por sobre la iluminación difusa que aparece dispersa bajo la superficie.

front_sss_color **Front surface scatter color** y **front_sss_weight** **Front surface scatter weight** son respectivamente el color y la influencia de la dispersión lumínica frontal de la superficie.

front_sss_radius **Front surface scatter radius** es el radio de dispersión frontal de la superficie. La luz se dispersará a través de la superficie, la distancia aquí indicada (en las unidades en que se esté trabajando) dividida por el parámetro *scale_conversion*.

back_sss_color **Back surface scatter color** y **back_sss_weight** **Back surface scatter weight** son respectivamente el color y la influencia de la dispersión lumínica trasera. En otras palabras: la luz que atraviesa al objeto.

back_sss_radius **Back surface scatter radius** y **back_sss_depth** **Back surface scatter depth** son respectivamente el radio de dispersión y la profundidad hasta la cual la luz proveniente de la parte trasera es visible en la superficie delantera del objeto. Normalmente el radio y la profundidad se fijan en el mismo valor (y en caso de no estar especificada, la profundidad toma el valor del radio) pero pueden establecerse por separado para obtener un mayor control.

specular **Specular color** es el color y **exponent** **Shininess** la intensidad de los brillos especulares, de tipo Phong.

lightmap_gamma **Lightmap gamma curve** es la curva gama de la luz almacenada en el mapa de iluminación. Si es 1, se almacena una iluminación difusa normal de tipo Lambert. Si es menor que 1, la curva es 'aplanada' causando que la luz se esparza hacia áreas perpendiculares al ángulo de incidencia de la luz. Si el valor es mayor que 1, la curva es 'ahusada' causando que la luz se concentre en las áreas que enfrentan directamente a la fuente lumínica. En la mayoría de los casos, un valor de entre 0,4 y 0,8 será apropiado.

indirect **Scatter indirect illumination**, si es verdadero, el mapa de iluminación incluirá a la iluminación indirecta (fotones, final gathering, etc) para ser dispersada. Esto tomará tiempo adicional en el render y se debe juzgar caso a caso si la contribución de la iluminación indirecta al efecto de dispersión de la luz es suficientemente significativa o no como para justificar el tiempo extra de cálculo.

scale_conversion **Scale conversion factor** es una sencilla función utilitaria que permite una división lineal de todas las distancias. Como la dispersión es dependiente de la distancia, asignar un material diseñado para un modelo hecho en pulgadas no funcionará correctamente en uno hecho en metros y viceversa.. Aquí se especifica el factor de conversión.

Cuando **scatter_bias** **Scatter Bias** es 0 la dispersión es completamente uniforme. Valores positivos favorecen la dispersión frontal, mientras que los negativos favorecen la dispersión trasera. El rango permitido es -1,0 a 1,0, pero el rango útil es mucho más pequeño. Un valor de 0,1 (una ligera influencia de la dispersión frontal) es un buen punto de partida.

falloff **Falloff strenght** establece la forma del desvanecimiento de la intensidad a lo largo del radio de dispersión. Valores altos obtienen un desvanecimiento más pronunciado, mientras que valores más pequeños obtienen uno más suave, pero también hacen parecer que la distancia de dispersión es más pequeña, de manera que se debe compensar esto incrementando la propia distancia de dispersión para una apariencia ligeramente más 'suave'. Con valores altos (1,0 a 10,0), casi todas las muestras dentro del radio de dispersión tienen una influencia similar. Con valores bajos (0,1 a 1,0), las muestras próximas al borde del radio de dispersión tienen menos influencia. ??? pa mi que es alverre!!!

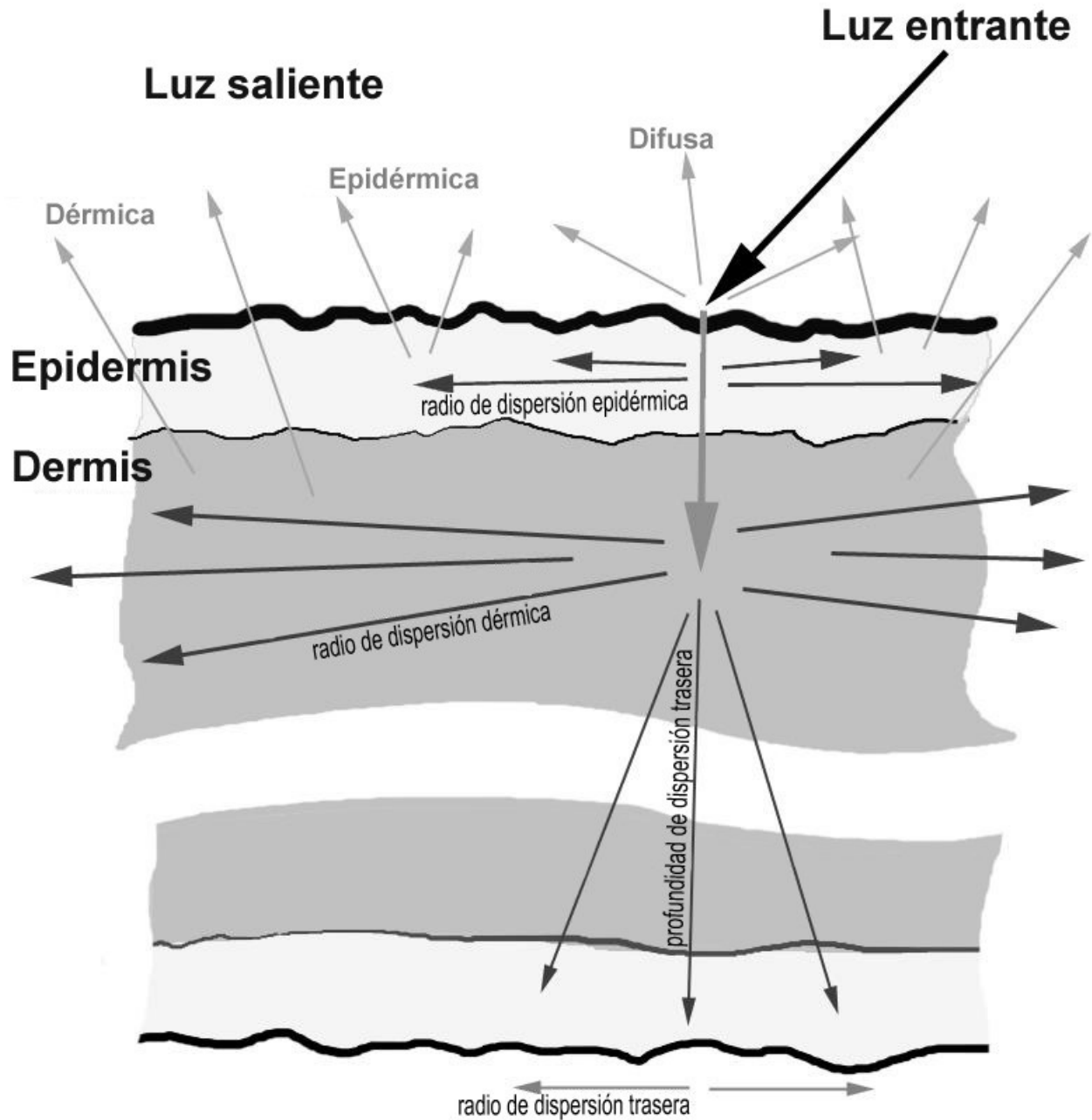
screen_composit **'Screen' compositing of layers**, cuando está activada, se realiza una composición de tipo "Screen". Cuando simplemente se suman las contribuciones de las varias capas de luz es muy fácil que rápidamente se obtenga una sobre exposición y que el resultado quede "quemado". Sin embargo el ojo humano es intrínsecamente no lineal y percibe las intensidades de una forma diferente. Esta opción permite aplicar lo que en muchas aplicaciones de composición es conocido como un modo de transferencia de tipo "screen" entre las capas, lo que logra un resultado más suave y agradable a la vista. Si la opción se encuentra desactivada se utilizará un modo aditivo simple para componer las distintas capas de luz. Si se realiza un render con alto rango dinámico y se aplica un control de exposición apropiado antes de la salida para comprimir la luminancia final, entonces es conveniente dejar desactivada esta opción. En un render normal, al activar esta opción se obtendrá un resultado más agradable.

mode es el selector de modo para la lista de luces: todas (0), inclusivo (1) o exclusivo (2). "Todas" es más rápido que "inclusivo", e "inclusivo" es más rápido que "exclusivo".

lights es la lista de luces.

misss_fast_skin_phen **SSS Fast Skin Material (mi)**

Un Fenómeno más avanzado para sombreado de piel que incluye tres capas de dispersión y funcionalidad de especularidad específica para la piel. Comparte varios parámetros con *misss_fast_simple_phen*.



Estructura del shader de piel

La piel es simulada mediante capas, tal como se detalla: Una capa reflectiva de lustre (para obtener especularidad y reflectividad difusa superficial no dispersa de tipo Lambert), una dispersión por debajo de la superficie de dos niveles (simulando tanto la dispersión en la epidermis como en las capas más profundas, referidas aquí colectivamente como “dérmica”) y una dispersión “trasera” que atraviesa la piel desde el lado opuesto.

```

material "misss_fast_skin_phen" (
    color texture "lightmap",
    color texture "depthmap",
    string        "lightmap_group",
    scalar        "lightmap_size",
    integer       "samples",
    shader        "bump",
    struct "d" {
        color "ambient",
        color "overall_color",
        color "diffuse_color",
        scalar "diffuse_weight",

        color "front_sss_color",
        scalar "front_sss_weight",
        scalar "front_sss_radius",
        color "mid_sss_color",
        scalar "mid_sss_weight",
        scalar "mid_sss_radius",
        color "back_sss_color",
        scalar "back_sss_weight",
        scalar "back_sss_radius",
        scalar "back_sss_depth"
    },
    struct "s" {
        scalar "overall_weight",
        scalar "edge_factor",
        color  "primary_spec_color",
        scalar "primary_weight",
        scalar "primary_edge_weight",
        scalar "primary_shininess",

        color  "secondary_spec_color",
        scalar "secondary_weight",
        scalar "secondary_edge_weight",
        scalar "secondary_shininess",

        scalar "reflect_weight",
        scalar "reflect_edge_weight",
        scalar "reflect_shininess",
        boolean "reflect_environment_only",
        shader "environment"
    },
    struct "a" {
        scalar "lightmap_gamma",
        boolean "indirect",
        scalar "scale_conversion",
        scalar "scatter_bias",
        scalar "falloff",
        boolean "screen_composit"
    },
    integer "mode", # light selection mode 0..2
    array light "lights"
)

```

Los parámetros **lightmap**, **depthmap**, **lightmap_group**, **Scatter group**, **lightmap_size**, **Lightmap size**, **samples**, **Number of samples**, **bump**, **Bump shader**, **ambient**, **Ambient / Extra light**, **lightmap_gamma**, **Lightmap gamma curve**, **indirect**, **Scatter indirect illumination**, **scale_conversion**, **Scale conversion factor**, **scatter_bias**, **Scatter Bias**, **falloff**, **Falloff strenght** y **screen_composit**, **'Screen' compositing of layers** funcionan de forma idéntica a sus contrapartes en el shader *missss_fast_simple_phen*.

overall_color, **Overall diffuse coloration** es la coloración global de todos los componentes difusos. Este color afectará a todas las capas por debajo (no dispersa, epidérmica, dérmica, trasera) y es útil para retoques globales de color o para mapear características de la superficie de la piel que deberían bloquear la luz proveniente del interior de la misma, tal como lunares, cejas que se representen mediante mapas, suciedad, etc.

diffuse_color, **Unscattered diffuse color** es el color y **diffuse_weight**, **Unscattered diffuse weight** es la influencia del componente difuso normal, de tipo Lambert. Esto representa a la capa más externa de la piel, donde aparecen todas las pequeñas rugosidades. Para obtener una piel de tipo caucásica el color debería ser mayormente blanco con, quizás, un pequeño componente azul. Un valor apropiado de influencia sería de alrededor de 0,3.

front_sss_color, **Epidermal layer scatter color**, **front_sss_weight**, **Epidermal layer scatter weight** y **front_sss_radius**, **Epidermal layer scatter radius** son respectivamente el color, influencia y radio de dispersión de la capa epidérmica de la piel, que es la capa blancuzca/amarillenta justo por debajo de la superficie. Buenos valores iniciales son un color ligeramente amarillento, una influencia de 0,5, y un radio de dispersión de alrededor de 5 a 10mm (0,2 a 0,4 pulgadas).

mid_sss_color, **Subdermal layer scatter color**, **mid_sss_weight**, **Subdermal layer scatter weight** y **mid_sss_radius**, **Subdermal layer scatter radius** son respectivamente el color, influencia y radio de dispersión de la capa dérmica de la piel. La funcionalidad de los parámetros es la misma que la de los que comienzan con "front_" pero esta vez aplicado a una capa que se encuentra por debajo de las superiores y que representa a la dispersión más profunda y rojiza, "color carne". Un color rojizo/anaranjado con una influencia de alrededor de 0,5 y un radio de dispersión 10 a 25mm (0,4 a 1,0 pulgadas) es un buen punto de partida para la experimentación.

back_sss_color, **Back surface scatter color**, **back_sss_weight**, **Back surface scatter weight**, **back_sss_radius**, **Back surface scatter radius** y **back_sss_depth**, **Back surface scatter depth** representan a la luz que pasa a través de la carne (por ejemplo para lograr apariencia de "orejas rojas").

overall_weight, **Overall specular Weight** es el nivel global de especularidad y reflectividad. Generalmente el mapa de especularidad se usa aquí, afectando el nivel de las restantes opciones de especularidad que le siguen debajo.

edge_factor, **Edge narrowness** establece el grosor del borde para los efectos de reflectividad en bordes. La piel se ve más reflectiva cuando es vista en ángulos sesgados (esto es conocido como "efecto Fresnel"). Este parámetro establece en efecto qué tan angosto es este borde. Valores más altos provocarán un borde más angosto. El ancho de este borde afectará a todos los valores de influencia de borde que aparecen debajo.

primary_spec_color, **Specular Color #1** y **primary_spec_weight**, **Specular Weight #1** establecen el color y la influencia de la primera capa de brillo especular. La especularidad de la piel es tratada en dos capas, permitiendo la simulación tanto de la especularidad suave y amplia de la piel, como de otras casi reflectivas de capas superiores aceitosas o mojadas.

primary_edge_weight, **Specular Edge Weight #1** establece un multiplicador adicional

para la influencia de la especularidad en los bordes, donde la especularidad resultante será la suma de **weight** y **edge weight**.

primary_shininess **Shininess #1** es el exponente de especularidad (valores mayores logran un brillo especular más pequeño y definido, derivado del modelo Phong con un suavizado de borde).

secondary_spec_color **Specular Color #2**, **secondary_spec_weight** **Specular Weight #2**, **secondary_edge_weight** **Specular Edge Weight #2** y **secondary_shininess** **Shininess #2** funcionan exactamente igual que los parámetros que comienzan con "primary_" y controlan la segunda capa de especularidad. Para lograr piel, una buena guía básica es tener una primera capa de especularidad con un valor de *shininess* muy bajo (3,0 a 8,0), un nivel bajo de *weight* (0,1 a 0,3), pero con un valor de *edge weight* bastante alto (0,5 a 1,0) y un color ligeramente azulado; y una segunda capa de especularidad con un *shininess* alto (20 a 100), un *weight* medio (0,3 a 0,6) y sin una influencia extra en los bordes (*edge_weight*). Introducir un mapa en los canales de especularidad realza en gran medida el realismo obtenido.

reflect_weight **Reflection Weight** es la influencia y **reflect_edge_weight** **Reflection Edge Weight** la influencia en el borde para los reflejos de la piel. Si estos valores son mayores a cero, se agregará reflectividad a la piel.

reflect_shininess **Reflection glossiness** es el grado de difusión de los reflejos sobre la piel. Cuando el valor es cero (0,0), se utiliza el método raytrace tradicional. Para valores distintos de cero se generan reflejos difusos, lo que incrementa el tiempo de cálculo.

Si **reflect_environment_only** **Only reflect environment** se encuentra activado, sólo se reflejará el mapa de entorno actualmente asignado, no trazándose ningún rayo.

environment **Local environment** especifica un mapa de entorno local (sólo para ese material).

mode es el selector de modo para la lista de luces: todas (0), inclusivo (1) o exclusivo (2). "Todas" es más rápido que "inclusivo", e "inclusivo" es más rápido que "exclusivo".

lights es la lista de luces.

misss_fast_skin_phen_d **SSS Fast Skin Material+Displace (mi)**

Este fenómeno es un duplicado exacto del *misss_fast_skin_phen*, pero que soporta mapeo de desplazamiento. Este debe ser un Fenómeno separado, puesto que un Fenómeno con desplazamiento siempre intentará desplazar al objeto.

```
=> material "misss_fast_skin_phen_d" (  
    color texture "lightmap",  
    color texture "depthmap",  
    string        "lightmap_group",  
    scalar        "lightmap_size",  
    integer       "samples",  
    shader        "displace",  
    shader        "bump",  
    struct "d" {  
        ...  
    }  
    integer       "mode",          # light selection mode 0..2  
    array light   "lights"  
)
```

displace **Displacement** es un shader que desplaza la superficie.

Todos los demás parámetros son idénticos a los de *misss_fast_skin_phen*.