

## Пример использования поляризационной камеры для получения нормалей к поверхности при анализе глубины объекта

В этой статье описан процесс создания трёхмерной модели объекта, полученной на основе нормалей к поверхности с помощью поляризационной камеры. Для анализа параметров линейной поляризации (степени (DoLP – Degree of Linear Polarization), угла (AoLP – Angle of Linear Polarization) поляризации и интенсивности) света, отражённого от объекта, получаемые из изображения с поляризационных камер LUCID Phoenix и Triton и вычисленных на их основе нормалей к поверхности объекта.

Нормали к поверхности, которые находятся за пределами некоторого допуска или сильно отличаются от соседних нормалей, могут рассматриваться как дефекты и деформации (например, вмятины на металлической банке).

Кроме того, нормали к поверхности также можно использовать для восстановления трехмерного облака точек объекта. Этот метод полезен при анализе качества формы объекта, если у вас нет доступа к 3D-камере глубины (ToF).

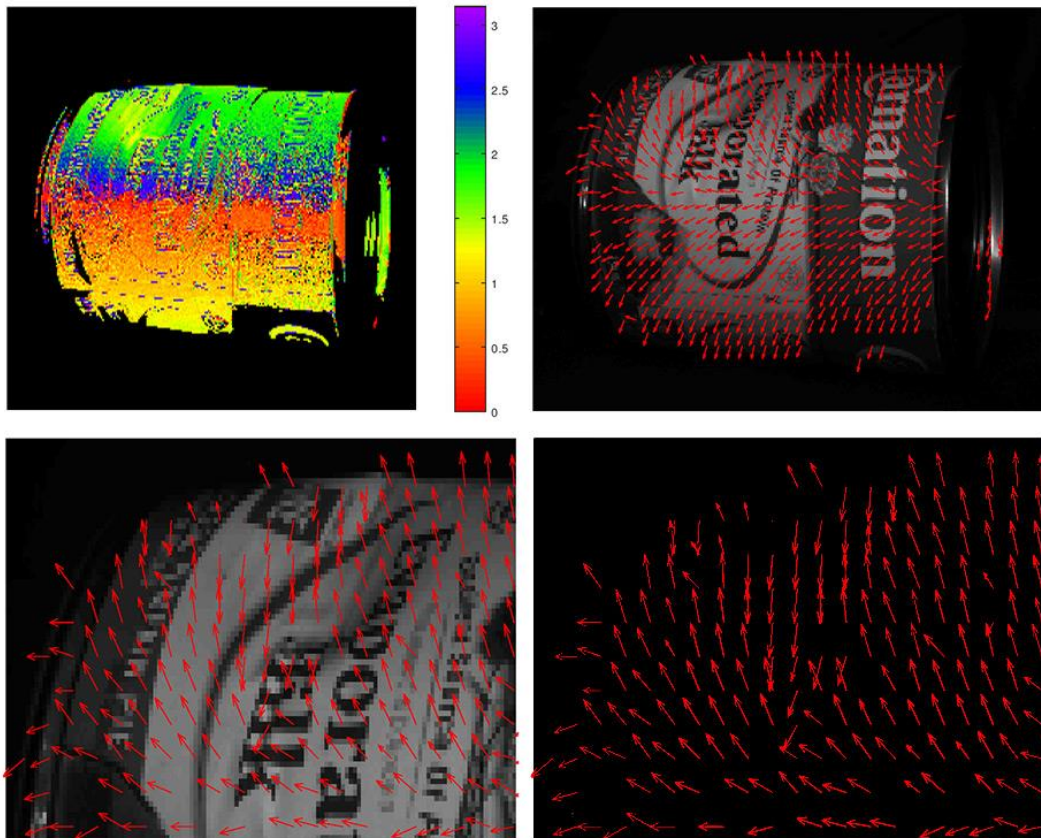
Оборудование и ПО:

- Камера технического зрения Lucid Triton ([TRIO50S-P/Q](#)) или Phoenix ([PHX050S-P/Q](#)) 5.0 МП с поляризационным сенсором;
- Объектив Fujinon C-Mount SMP 2/3" 8mm f/1.6;
- Источник неполяризованного света (SCHOTT EasyLED Spot Light, [Edmund Optics #15-921](#));
- Программное обеспечение Arena SDK (ArenaView для захвата изображений);
- Matlab или GNU Octave.

## Деформация на примере помятой металлической банки

### С деформацией

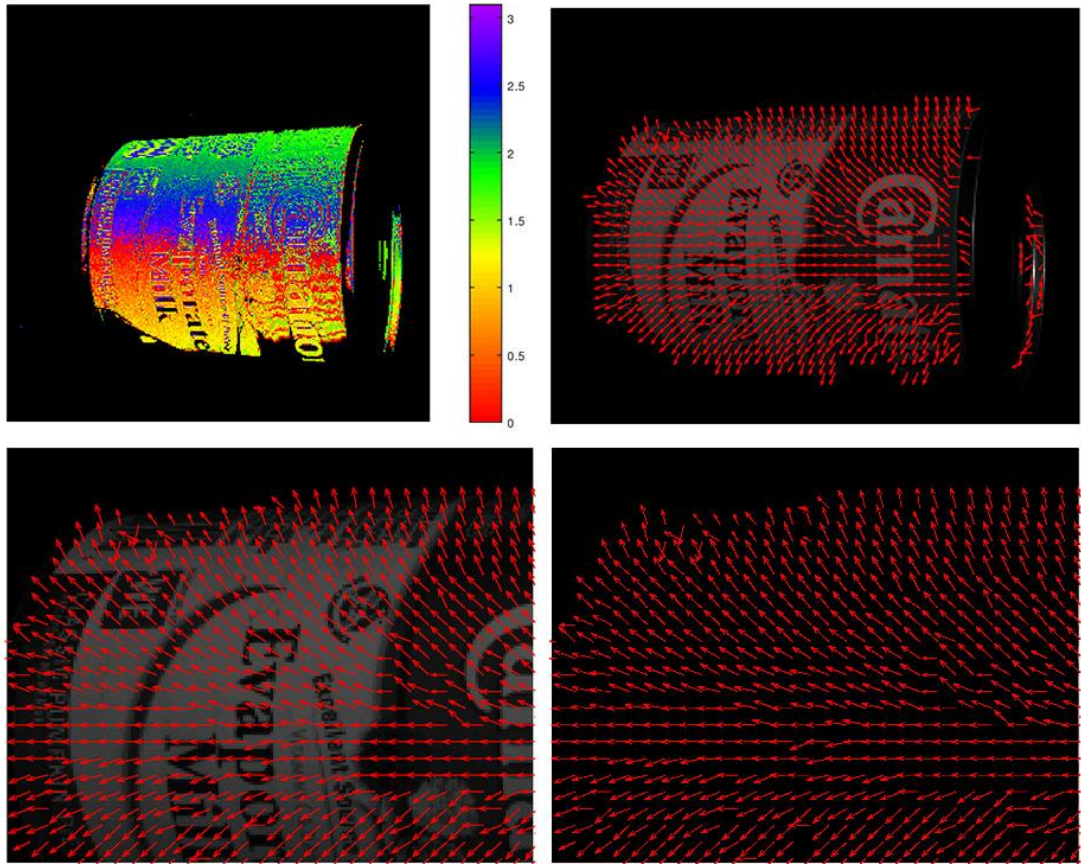
На изображениях ниже показано, как нормали к поверхности (красные стрелки) могут использоваться для определения деформации формы объекта. Вмятину можно увидеть, если нормали к поверхности направлены в стороны, отличные от окружающих нормалей.



Изображение помятой банки с использованием поляризационной камеры Triton

### Без деформации

В банке без какой-либо деформации, гладкая поверхность создает более согласованные нормали поверхности, направленные в одном направлении. Как в примерах с вмятинами, так и в примерах без вмятин, мы можем восстановить карту глубины для создания трехмерного облака точек.



Изображение целой банки с использованием поляризационной камеры Triton

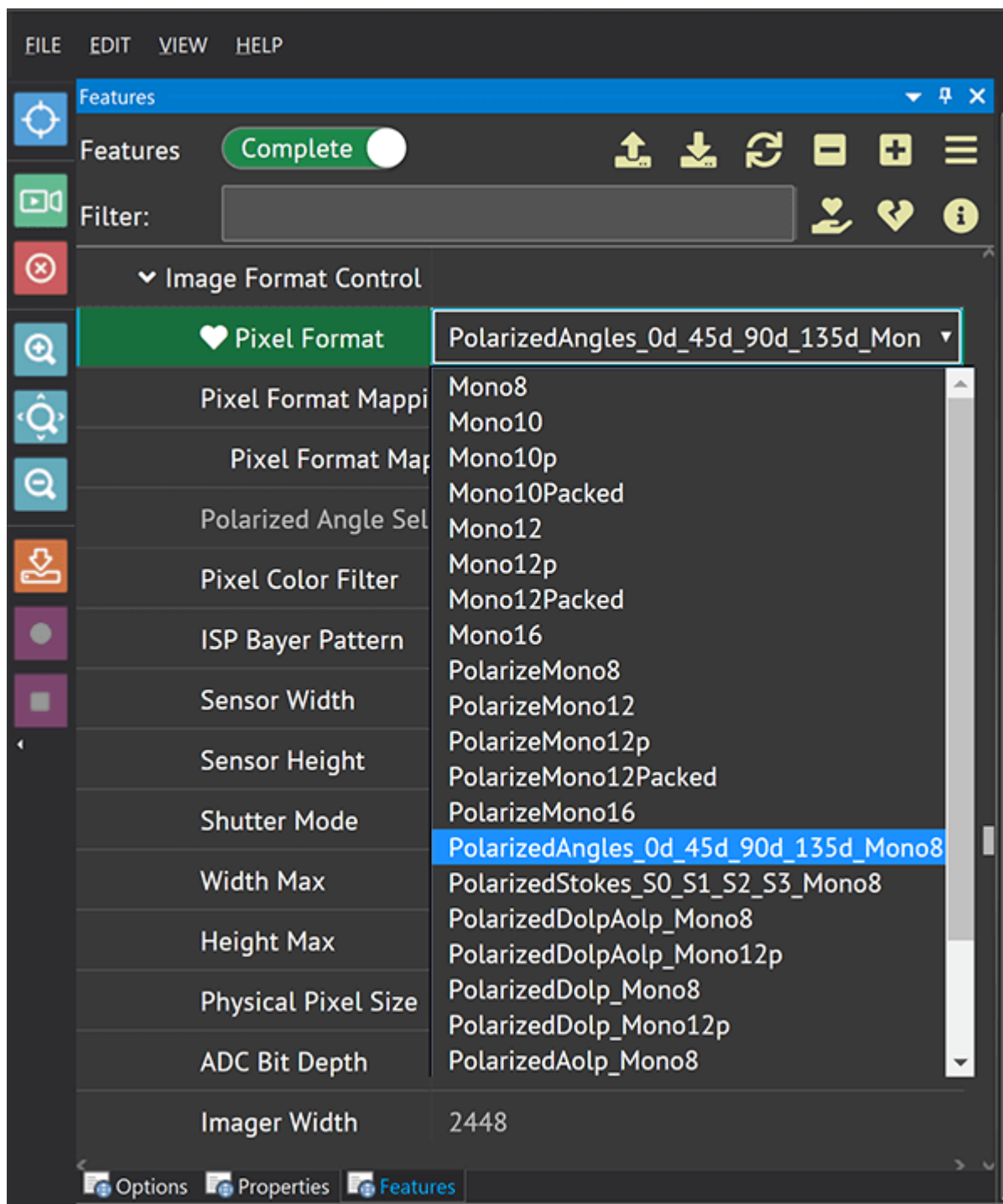
## Пример настроек и кода

### Шаг 1. Установка камеры, света и объекта

Установка камеры, источника света и объекта съёмки должна обеспечивать оптимальные углы. Это достигается экспериментально, но, в целом, осветитель должен располагаться рядом с объектом (примерно 45 градусов к камере). Также, чтобы было легче фильтровать фон, лучше использовать однотонный цветной или тёмный фон.

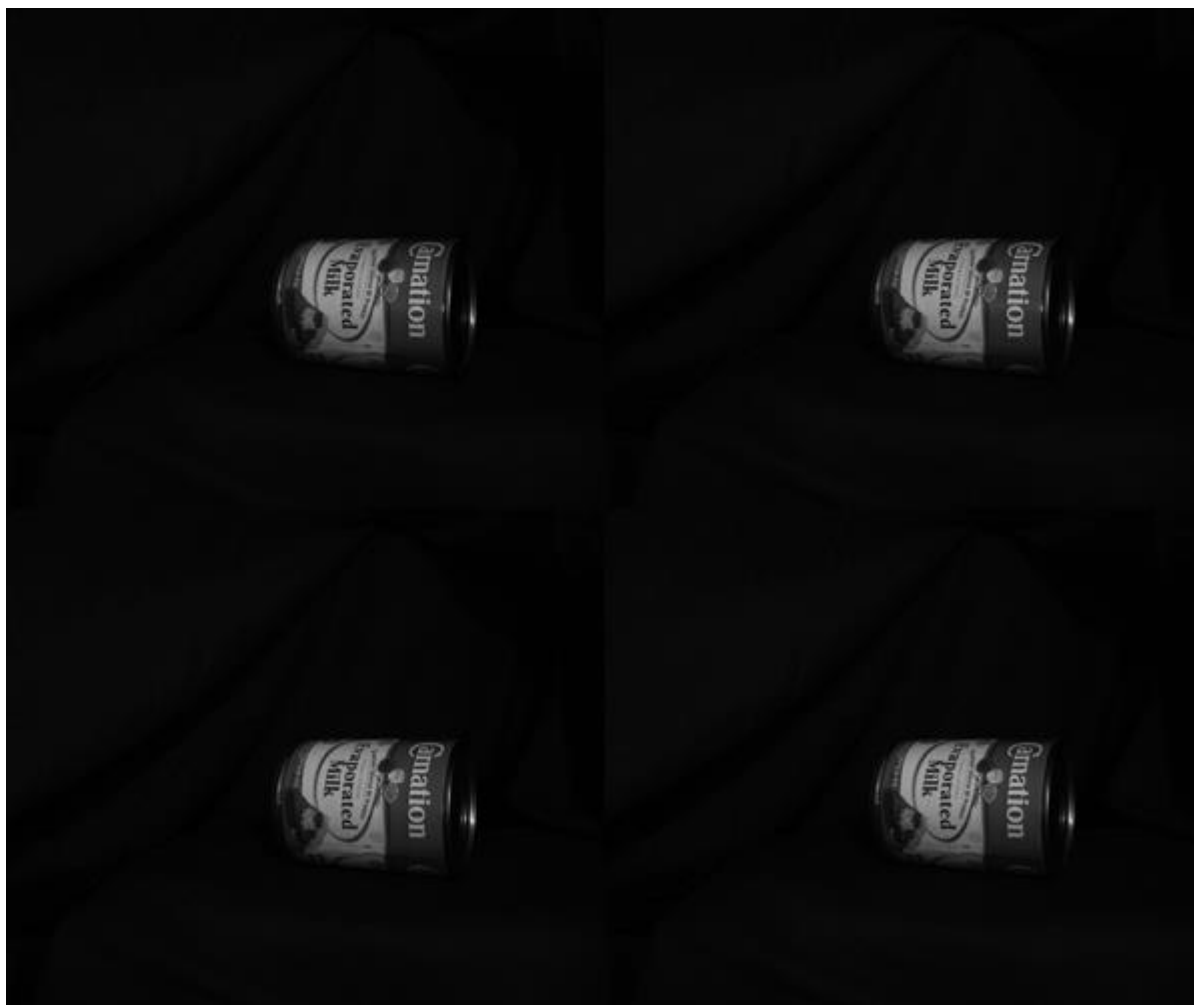
### Шаг 2. Установка атрибута pixel format в значение "0°, 45°, 90° and 135° polarized channels"

Подключите камеру Triton или Phoenix к ПК и запустите ArenaView. Выберите атрибут "PolarizedAngles\_0d\_45d\_90d\_135d\_Mono8". Далее, начните захват изображений.



Полученное изображение будет использовано для оценки нормалей поверхности.

Ваше изображение должно быть похоже на изображение ниже, с 4 фотографиями в одной картинке (каждый кадр отображает один из поляризационных углов).



Пример изображений, полученных с PolarizedAngles\_0d\_45d\_90d\_135d\_Mono8 pixel format.

### Шаг 3. Скачайте исходный код и запустите в Matlab или Octave

[Скачать Code.zip \(6.0 MB\)](#)

Name	Size	Date modified	Type
data		2021-01-04 3:54 PM	File folder
DCT_Poisson.m	2 KB	2021-01-04 2:57 PM	M File
dct2.m	3 KB	2011-10-11 9:23 AM	M File
demo_depth_from_polarization.m	3 KB	2021-01-04 3:16 PM	M File
idct2.m	3 KB	2011-10-11 9:23 AM	M File
lookup_aolp_cylinder.m	1 KB	2021-01-04 2:57 PM	M File
PolarisationImage.m	3 KB	2021-01-04 2:57 PM	M File

Содержание архива Code.zip

Запустите файл **demo\_depth\_from\_polarization.m**. Обратите внимание, что существует несколько методов определения нормалей к поверхности. В

приведённом коде, для оценки AoLP, используется подход "look-up table" (таблица поиска). Данный подход работает только для цилиндрических объектов, однако другие методы описаны вместе с кодом на [github](#).

```
clear all

close all

%% Load the captured image of four polarization angles

in_path = 'data/milk_can.tiff';

im = imread(in_path);

%% Stack the image of four polarization angles into an image of four channels

[nrows, ncols] = size(im);

im0 = im(1 : nrows/2, 1 : ncols/2);

im45 = im(1 : nrows/2, ncols/2 + 1 : end);

im90 = im(nrows/2 + 1 : end, 1 : ncols/2);

im135 = im(nrows/2 + 1 : end, ncols/2 + 1 : end);

images = zeros(nrows/2, ncols/2, 4);

images(:, :, 1) = double(im0);

images(:, :, 2) = double(im45);

images(:, :, 3) = double(im90);

images(:, :, 4) = double(im135);

nskips = 4; % may work with a smaller image which runs faster
```



```

images = images(1 : nskips : end, 1 : nskips: end, :);

%% Assign polarization angles to corresponding channels

angles = [0, 45, 90, 135] * pi / 180;

%% May create a mask for a region of interest

use_fg_threshold = true;

mask = ones(size(images(:, :, 1)));

if ( use_fg_threshold )

    image_avg = mean(double(images), 3);

    fg_threshold = 10;

    mask(image_avg < fg_threshold) = 0;

    mask(image_avg >=fg_threshold) = 1; % foreground

end

mask = logical(mask);

figure; imagesc(mask)

%% Calculate polarization attributes of the image: dola, aola and intensity

% credit: PolarizationImage can be found from https://github.com/waps101/depth-from-polarisation

[ dola_est,aola_est,intensity_est ] = PolarisationImage( images,angles,mask,'linear' );

mask(dola_est < 0.005) = 0; % filter out pixels with very small degree of polarization

figure; imagesc(aola_est); colorbar; colormap (rainbow)

figure; imagesc(dola_est); colorbar

```

```

figure; imagesc(intensity_est); colormap gray

%% Estimate surface normals of the object

% This can done with

% 1. a Lambertian model; OR

% 2. a boundary propagation method; OR

% 3. a simple look up table approach given the lighting setup is invariant

% The code of methods 1 and 2 can be found in https://github.com/waps101/depth-from-polarisation

N = lookup_aolp_cylinder(aolp_est);

%[N, height] = Propagation( rho_est,phi_est,mask,n );

% We found that a median filter could be used to reduce the noise of the estimated normal vector but it is not a must

N(:,:,1) = imsmooth(N(:,:,1), "Median", [5,5]);

N(:,:,2) = imsmooth(N(:,:,2), "Median", [5,5]);

N(:,:,3) = imsmooth(N(:,:,3), "Median", [5,5]);

%% Depth reconstruction from the derived surface normals

% Various methods are described and their codes are included in https://github.com/yqueau/normal\_integration

temp = N(:, :, 3);

temp(abs(temp)<1e-5) = nan; % avoid dividing by zero or a very small number. otherwise it will screw up the depth map

N(:, :, 3) = temp;

P = -N(:,:,1)./N(:,:,3);

```

```

Q = -N(:,:,2)./N(:,:,3);

P(isnan(P)) = 0;

Q(isnan(Q)) = 0;

height = DCT_Poisson(P,Q);

figure; imagesc(height); colorbar

height(~mask) = nan;

figure;

surf(height,'EdgeColor','none','FaceColor',[0 0 1],'FaceLighting','gouraud','AmbientStrength',0,'DiffuseStrength',1);

axis equal; light

xlabel('x')

ylabel('y')

zlabel('z')

```

## Заключение

Анализ информации о поляризации может быть очень полезен при проверке геометрической формы объекта без необходимости использования традиционной 3D-камеры. Благодаря камерам Phoenix и Triton с поляризацией, пользователи могут получить такие параметры, как AoLP, DoLP и интенсивность поляризации, благодаря свету, отраженному от объекта, для оценки нормалей к поверхности, которые затем проверяются или используются для восстановления глубины. Деформации объекта могут быть выявлены по отклонению некоторых нормалей от направления соседних. В примере, описанном выше, данные о AoLP металлической банки отображаются в Look-Up-Table, предназначенной для оценки нормалей поверхности цилиндрических объектов. Также, разные таблицы LUT могут быть использованы для объектов разных форм и пользователи могут найти на Github и другие методы. Наконец, данные поляризации могут быть использованы для воссоздания трёхмерного облака точек объекта (Форма поляризации). Этот пример показывает, как возможно успешно инспектировать свойства трёхмерного объекта, используя двухмерную поляризационную камеру.