UNIVERSIDAD NACIONAL DE SAN AGUSTÍN
ESCUELA DE POSTGRADO
UNIDAD DE POSTGRADO DE LA FACULTAD DE INGENIERÍA DE PRODUCCIÓN Y SERVICIOS
DOCTORADO EN CIENCIAS DE LA COMPUTACIÓN

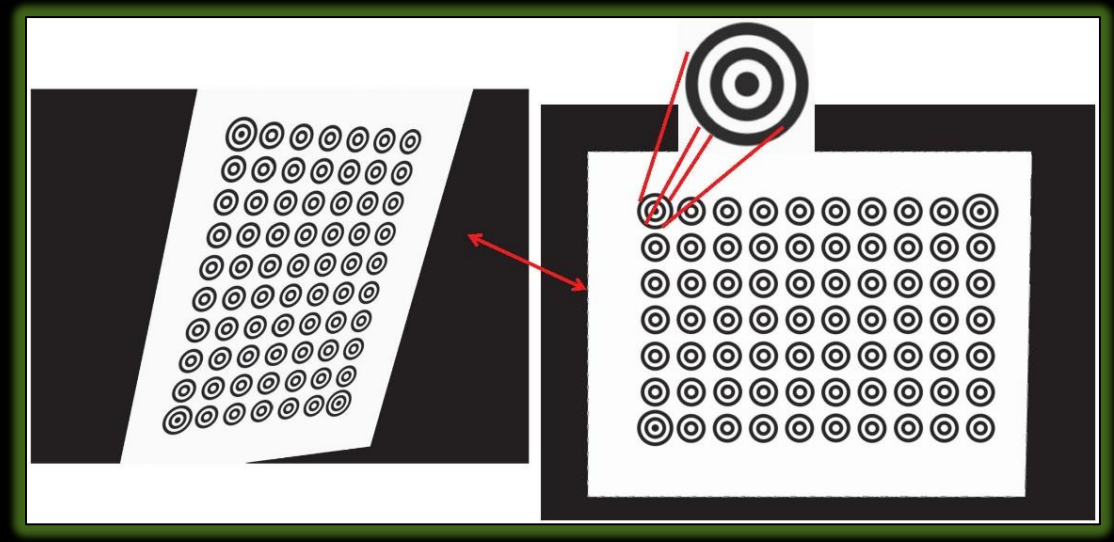
Calibración de Cámara Basado en un Patrón de Anillos

Presentado por:
Ing. Leticia Laura Ochoa
Ing. Maribel Molina Barriga

Arequipa – Perú
2018

CALIBRACION DE CAMARA

- Consiste en obtener los parámetros intrínsecos y extrínsecos que definen el modelo de la cámara.
- Existen varios métodos de calibración que utilizan plantillas planas de cuadrículas, círculos y anillos.
- En este trabajo se investigará los pasos para la calibración de cámara basado en el patrón de anillos.



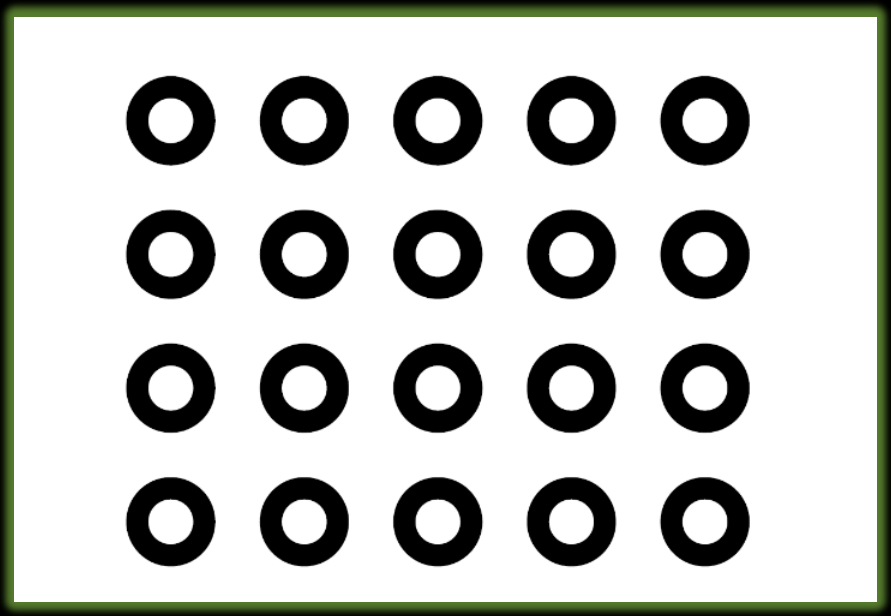
ABRIENDO LA CAMARA CON EL VISUAL STUDIO PARA CAPTURAR IMÁGENES

```
#include<opencv\cv.h>
#include<opencv\highgui.h>
using namespace cv;
int main()
{
	Mat image; //Crear una matriz para almacenar la imagen
	VideoCapture cap; //Inicializar la captura
	cap.open(0);

	namedWindow("Windows",1); //crear la ventana para mostrarla imagen
	while(1)
	{
		//copiar la secuencia de la cámara web a la imagen
		cap>>image;
		//imprimir imagen en pantalla
		imshow("window",image);
		//retrasar 33 ms
		waitKey(33);
	}
	return 0;
}
```

PREPARACIÓN DEL PATRÓN

- Preparación de los patrones: Prepare un patrón plano de pares de círculos concéntricos. La posición del círculo centro y el radio de cada círculo es conocido. Al menos se necesitan dos pares, tome imágenes de este patrón plano, al menos 7 imágenes son necesarias.



CAPTURA DE FRAMES A PARTIR DE UN VIDEO

```
int main()
{
	std::vector<int> compression_params;
	compression_params.push_back(CV_IMWRITE_JPEG_QUALITY);
	compression_params.push_back(100);
	Mat imagen;
	VideoCapture video("PadronAnillos_02.avi");
	int count = video.get(CV_CAP_PROP_FRAME_COUNT);
	bool guardarFrame;
	std::string filePath = "E:\\FramesVideo\\Frame";
	std::stringstream saveFrame;
	for (int frameNum = 0; frameNum < count ; frameNum++)
	{
		video >> imagen;
		namedWindow("Video", 1);
		imshow("Video", imagen);
		saveFrame << filePath << frameNum << ".jpg";
		guardarFrame = imwrite(saveFrame.str().c_str(), imagen, compression_params);
		saveFrame = std::stringstream();
	}
	cvWaitKey();
	return 0;
}
```

PRE PROCESADO DE DATOS PARA LA CALIBRACIÓN