



# Desenvolvimento de um Sistema de Monitorização de Respiração para Pacientes de Radioterapia Externa

PAULO ANDRÉ NETO DA ROCHA SANTOS

Outubro de 2023

# DESENVOLVIMENTO DE UM SISTEMA DE MONITORIZAÇÃO DE RESPIRAÇÃO PARA PACIENTES DE RADIOTERAPIA EXTERNA

Paulo André Neto da Rocha Santos

2023

Instituto Superior de Engenharia do Porto

Departamento de Física

isen

P.PORTO

# DESENVOLVIMENTO DE UM SISTEMA DE MONITORIZAÇÃO DE RESPIRAÇÃO PARA PACIENTES DE RADIOTERAPIA EXTERNA

Paulo André Neto da Rocha Santos

Estudante n.º 1171018

Dissertação apresentada ao Instituto Superior de Engenharia do Porto para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia Biomédica, realizada sob a orientação do Prof. Dr. Luís Cunha e da Prof. Dra. Gina Vilão e coorientação do Mestre Bruno Mendes e Prof. Dr. Luiz Faria

**2023**

Instituto Superior de Engenharia do Porto

Departamento de Física

isen

P.PORTO

## **AGRADECIMENTOS**

Primeiramente, gostava de agradecer ao Instituto Superior de Engenharia do Porto pelos professores que me acompanharam durante estes dois anos de curso e contribuíram para o meu desenvolvimento pessoal e profissional.

De seguida, gostaria de agradecer aos meus orientadores com especial destaque ao Prof. Dr. Luís Cunha, do Instituto Português de Oncologia do Porto, pelo apoio incansável durante estes meses e à Prof. Dra. Gina Vilão pelo acompanhamento e dedicação desde o início do ano letivo e por me ter proporcionado a oportunidade de realizar este projeto. O mesmo foi realizado no IPO-Porto, cujo instituto eu agradeço por me ter recebido nas suas instalações e por toda a ajuda e carinho que me foi providenciado.

Gostaria também de agradecer ao Prof. Paulo Matos do Departamento de Informática do Instituto Superior de Engenharia do Porto pela disponibilização da impressora 3D.

Por último, mas não menos importante, gostava de agradecer à minha família e amigos por serem o meu porto de abrigo diário, a minha motivação e inspiração durante esta jornada de aprendizagem.

página propositadamente em branco

## RESUMO

Uma das opções mais recomendada e prescrita para o tratamento de cancro é a radioterapia. Este procedimento consiste em utilizar feixes de alta energia para matar células cancerígenas, sendo que o seu objetivo principal é destruir um menor número possível de células saudáveis, atingindo apenas os alvos que necessitem de radiação. Como forma de juntar os processos de radioterapia com os benefícios das novas tecnologias, criou-se um sistema de monitorização da respiração capaz de controlar e identificar os diferentes momentos de respiração de cada doente oncológico. Tendo como método de comparação o sistema da VARIAN®, o software desenvolvido tem como base a linguagem de programação *Python*, a utilização de um marcador físico, uma câmara para acompanhar os movimentos e uma calibração realizada no *background* de todo o sistema. O marcador físico revelou-se uma forma mais simples de simular o comportamento respiratório pois, após o seu reconhecimento e calibração previamente realizada, faz a ligação imediata com uma câmara e todos os seus movimentos são monitorizados. Estes marcadores estão conectados a uma base de dados que, por sua vez, apresentam diferentes bibliotecas. Estas que podem ser interpretadas e utilizadas pelo *Python*, uma linguagem *open source*, de fácil compreensão e disponível para vários sistemas operacionais. O software apresenta quatro componentes: visualização de imagem em tempo real, gráfico do movimento detetado, imagem para indicar autorização para iniciar radiação e um conjunto de botões responsáveis por diferentes ações. Com todas estas variantes, construiu-se um sistema simples, eficaz, com um design intuitivo e capaz de realizar todas as tarefas consideradas necessárias para controlar os momentos de respiração de um doente oncológico em sessões de radioterapia.

PALAVRAS-CHAVE: Inspiração profunda, Monitorização, Detecção, Calibração, Interface Gráfica, *Python*, *ArUco*.



página propositadamente em branco

## ABSTRACT

One of the most recommended and prescribed options for cancer treatment is radiotherapy. This procedure consists of using high-energy beams to kill cancer cells, which its main objective is to destroy as few healthy cells as possible, only reaching targets that require radiation. As a way of combining radiotherapy processes with the benefits of new technologies, a breathing monitoring system was created capable of controlling and identifying the different breathing moments of each cancer patient. Using the VARIAN® system as a comparison method, the software developed is based on the Python programming language, the use of a physical marker, a camera to monitor movements and a calibration method in the background of the entire system. The physical marker proved to be a simpler way of simulating respiratory behaviour because, after its recognition and previous calibration, it immediately connects to a camera and all its movements are monitored. These markers are connected to databases which, in turn, contain different libraries. These libraries can be interpreted and used by Python, an opensource and easy-to-understand language, available for various operating systems. The software has four components: real-time image visualization, graph of detected movement, image to indicate authorization to start radiation and a set of buttons responsible for different actions. With all these variants, a simple, effective system was built, with an intuitive design and capable of carrying out all the tasks considered necessary to control the breathing moments of a cancer patient during radiotherapy sessions.

**KEYWORDS:** Deep inspiration breath hold, Monitoring, Detection, Calibration, Graphical Interface, Python, ArUco.

página propositadamente em branco

## ÍNDICE

ÍNDICE DE FIGURAS.....	VII
ÍNDICE DE TABELAS .....	IX
LISTAS DE SIGLAS .....	XI
1. INTRODUÇÃO .....	13
1.1. Enquadramento e pertinência .....	13
1.2. Objetivos de investigação.....	14
1.3. Local de Desenvolvimento .....	15
1.4. Estrutura do trabalho .....	15
2. REVISÃO BIBLIOGRÁFICA.....	17
2.1. Sistema Respiratório .....	17
2.1.1. Anatomia do Sistema Respiratório .....	17
2.1.2. Fisiologia do Sistema Respiratório .....	18
2.2. Sistemas de Monitorização da Respiração.....	18
2.2.1. Metodologias e Sistemas de Monitorização da Respiração .....	18
2.2.2. Impacto da Radioterapia na Função Respiratória.....	23
2.2.3. Movimentos Internos e Externos na Zona Torácica .....	24
2.3. Processamento do Sinal Respiratório.....	25
3. MÉTODOS E APLICAÇÃO.....	29
3.1. Marcadores Físicos.....	29
3.2. Interface Gráfica.....	32
3.3. Calibração .....	33
4. RESULTADOS E DISCUSSÃO .....	34
4.1. Janela de Calibração .....	34
4.2. Software Principal .....	39
5. CONCLUSÃO .....	43
5.1. Conclusões finais .....	43
5.2. Limitações e investigação futura.....	43
REFERÊNCIAS BIBLIOGRÁFICAS .....	45
APÊNDICE A.....	47

página propositadamente em branco

## ÍNDICE DE FIGURAS

<b>Figura 1:</b> DIBH, controlo da amplitude .....	13
<b>Figura 2:</b> Inspiração Profunda .....	14
<b>Figura 3:</b> Respiração Livre.....	14
<b>Figura 4:</b> Anatomia do Sistema Respiratório.....	17
<b>Figura 5:</b> Sistema ABC.....	19
<b>Figura 6:</b> VARIAN® RPM .....	19
<b>Figura 7:</b> Imagem de Raios-X ( <i>ExacTrac Dynamic</i> ).....	21
<b>Figura 8:</b> <i>Philips Bellows</i> .....	21
<b>Figura 9:</b> <i>Brilliance CT Big Bore</i> .....	22
<b>Figura 10:</b> Características de pneumonite por radiação .....	23
<b>Figura 11:</b> Gráficos da respiração de um paciente do estudo de 2009 .....	26
<b>Figura 12:</b> Marcador <i>ArUco</i> .....	31
<b>Figura 13:</b> Desenho no <i>SolidWorks</i> .....	31
<b>Figura 14:</b> Objeto impresso.....	32
<b>Figura 15:</b> Fase 1 da Calibração .....	34
<b>Figura 16:</b> Dados da Fase 2 da Calibração .....	35
<b>Figura 17:</b> Fase 3 da Calibração .....	36
<b>Figura 18:</b> Teste da Calibração.....	37
<b>Figura 19:</b> Janela da Lista dos Pacientes.....	40
<b>Figura 20:</b> Janela para Adicionar Novo Paciente.....	40
<b>Figura 21:</b> Janela de Monitorização da Respiração .....	41
<b>Figura 22:</b> Código Principal Parte 1.....	47
<b>Figura 23:</b> Código Principal Parte 2.....	48
<b>Figura 24:</b> Código Principal Parte 3.....	48
<b>Figura 25:</b> Código Principal Parte 4.....	49
<b>Figura 26:</b> Código Principal Parte 5.....	50
<b>Figura 27:</b> Código Principal Parte 6.....	51
<b>Figura 28:</b> Código Principal Parte 7.....	52
<b>Figura 29:</b> Código Principal Parte 8.....	53
<b>Figura 30:</b> Código Principal Parte 9.....	54
<b>Figura 31:</b> Código Principal Parte 10.....	55
<b>Figura 32:</b> Código Principal Parte 11.....	56
<b>Figura 33:</b> Código Principal Parte 12.....	57
<b>Figura 34:</b> Código Principal Parte 13.....	57
<b>Figura 35:</b> Código Principal Parte 14.....	58
<b>Figura 36:</b> Código Principal Parte 15.....	58
<b>Figura 37:</b> Código Principal Parte 16.....	59
<b>Figura 38:</b> Código Principal Parte 17.....	59
<b>Figura 39:</b> Código da Calibração Parte 1.....	60
<b>Figura 40:</b> Código da Calibração Parte 2.....	61
<b>Figura 41:</b> Código da Calibração Parte 3.....	62
<b>Figura 42:</b> Código da Calibração Parte 4.....	63
<b>Figura 43:</b> Código da Calibração Parte 5.....	64
<b>Figura 44:</b> Código da Calibração Parte 6.....	64
<b>Figura 45:</b> Código da Calibração Parte 7.....	65
<b>Figura 46:</b> Código da Calibração Parte 8.....	65

página propositadamente em branco

## ÍNDICE DE TABELAS

<b>Tabela 1:</b> Coordenadas $x$ do marcador.....	37
<b>Tabela 2:</b> Coordenadas $y$ do marcador .....	38
<b>Tabela 3:</b> Análise de erros na coordenada $x$ .....	38
<b>Tabela 4:</b> Diferenças de coordenadas do $y$ .....	39



página propositadamente em branco

## LISTAS DE SIGLAS

### Lista de Siglas

ISEP	Instituto Superior de Engenharia do Porto
P.Porto	Instituto Politécnico do Porto
IPO-Porto	Instituto Português de Oncologia do Porto
DIBH	<i>Deep Inspiration Breath Hold</i>
TAC	Tomografia Computorizada
SNC	Sistema Nervoso Central
PR	Pneumonite por Radiação
TCAR	Tomografia de Alta Resolução
CVF	Volume Máximo de Ar Exalado
VEF <sub>1</sub>	Volume de Ar Expirado no Primeiro Segundo
CPT	Capacidade Pulmonar Total
CI	Capacidade Inspiratória
DLCO	Difusão de Monóxido de Carbono
DC	Corrente Contínua
AC	Corrente Alternada
ABC	<i>Active Breath Control</i>
RPM	<i>Real-time Position Management</i>
2D	2 Dimensões
3D	3 Dimensões
4D	4 Dimensões
OBS	Número de Identificação de Doente Oncológico
EWMA	<i>Exponentially-Weighted Moving Average</i>

página propositadamente em branco

## 1. INTRODUÇÃO

A radioterapia é um tipo de tratamento de combate ao cancro que utiliza feixes de radiação eletromagnética de alta energia para matar células cancerígenas. Esta técnica danifica todo o género de células (quer as saudáveis, quer as cancerígenas), afetando o material genético que controla o crescimento e divisão celular. Embora afete todas as células existentes, o objetivo da radioterapia é destruir um número mínimo possível de células saudáveis. [1]

A tecnologia tem evoluído de forma progressiva ao longo dos últimos anos e, cada vez mais, a medicina vai necessitando das virtudes e oportunidades que esta área oferece, formando uma aliança forte, com objetivo de apresentar os melhores e mais práticos serviços de saúde, quer aos profissionais de saúde, quer aos utentes.

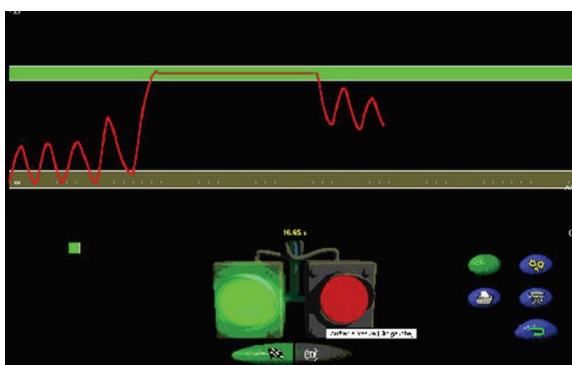
Assim, de forma a aliar o melhor que a radioterapia oferece com os benefícios da tecnologia, existem os sistemas de monitorização da respiração, com objetivo de haver um maior controlo e conhecimento do padrão de respiração de cada doente oncológico.

### 1.1. Enquadramento e pertinência

O trabalho enquadra-se na técnica de inspiração profunda denominada de DIBH (*Deep Inspiration Breath Hold*). Este género de técnica revela-se útil quando é necessário incidir radiação na zona torácica e abdominal, permitindo minimizar as doses na zona do coração ou pulmões e, assim, poupando aos órgãos danos desnecessários e possivelmente prejudiciais. [2]

Num processo padrão é pedido ao paciente para sustentar a respiração cerca de 20 segundos, sendo que o feixe de radiação só é ativado quando o doente se encontra em respiração profunda. Isto tudo porque, quando ocorre inspiração, o coração desloca-se da sua posição natural e afasta-se da zona de tratamento, havendo disponibilidade para incidir doses de radiação no local de interesse. [2] Da mesma forma, o volume de pulmão aumenta levando a uma diminuição da dose ao mesmo.

A figura 1 representa o padrão de respiração que se idealiza para este projeto, ilustrando a forma como a amplitude da respiração deve ser monitorizada.



**Figura 1:** DIBH, controlo da amplitude

O que verificamos na figura 1 é um exemplo de sistema que monitoriza a respiração do doente. Com esta técnica, é possível visualizar a evolução do volume pulmonar ao longo de um determinado período de tempo. Após umas sessões consideradas de treino, é estabelecida a sua amplitude de inspiração (zona verde da imagem). Inicialmente, o paciente está em respiração livre e, de seguida, começa o seu processo de inspiração profunda e é pedido para atingir a zona verde e manter-se entre esses limites. Enquanto

dura esta fase, incide-se radiação na zona que é necessário tratar e, após esse momento, o paciente expira e retoma à sua respiração livre, preparando-se para realizar as mesmas ações nos segundos seguintes. [2]

As figuras 2 e 3 representam os movimentos internos entre os órgãos em momentos de inpiração profunda e respiração livre, respetivamente.



**Figura 2:** Inspiração Profunda



**Figura 3:** Respiração Livre

Verifica-se, então, que quando o doente inspira fundo, o seu coração afasta-se ligeiramente da parede torácica e, portanto, mais distante de órgãos como a mama e o pulmão.

## 1.2. Objetivos de investigação

Os objetivos propostos para o projeto são os seguintes:

- Desenvolver um sistema de monitorização da respiração dos pacientes de modo a obter correlação entre o padrão respiratório e o movimento dos órgãos internos;
- Implementar/melhorar um algoritmo para deteção automática dos movimentos respiratórios do paciente;
- Permitir controlo da aquisição de imagens de planeamento e execução de tratamentos de radioterapia externa em função da respiração;
- Desenvolver interface gráfica para visualização do padrão respiratório por parte do paciente.

### **1.3. Local de Desenvolvimento**

Esta Tese de Mestrado desenvolveu-se em parceria com o Instituto Português de Oncologia do Porto, instituto que presta cuidados de saúde diferenciados no domínio da oncologia em Portugal.

### **1.4. Estrutura do trabalho**

O trabalho está dividido em cinco tópicos principais, a introdução, a revisão bibliográfica, métodos e aplicação, resultados e discussão e, conclusão. Antes de todos estes tópicos ainda é apresentado um resumo do trabalho.

Na introdução, são referidos os conceitos bases do trabalho, assim como, os objetivos propostos e o local de implementação onde foi realizado este projeto.

A revisão bibliográfica evidencia todos os factos teóricos e estudos relevantes para analisar e compreender da melhor forma o comportamento e funcionamento de um sistema de monitorização da respiração.

Na secção dos métodos e aplicação, são mostrados todos os procedimentos realizados ao longo dos meses, quais os métodos aplicados e como foram aplicados. Com base nos mesmos, obtiveram-se determinados resultados, evidenciados no tópico dos resultados e discussão.

A partir dos métodos e resultados, retiraram-se as conclusões principais do trabalho, sendo também mencionadas as limitações que apareceram ao longo de todo o desenvolvimento do projeto e sugestões para trabalhos futuros.



## 2. REVISÃO BIBLIOGRÁFICA

Neste capítulo, serão abordados temas relacionados com a monitorização da respiração e todos os subtemas que se consideraram relevantes e que tenham uma influência direta ou indireta com este género de sistemas responsáveis pelo controlo de momentos de respiração de um doente oncológico.

### 2.1. Sistema Respiratório

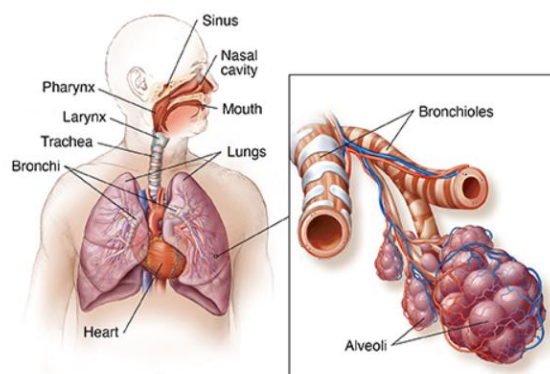
O sistema respiratório é constituído por órgãos e estruturas específicas que realizam trocas gasosas em animais e plantas. A anatomia e a fisiologia deste sistema são responsáveis por estes processos de troca, variando, de forma significativa, de acordo com o tamanho do organismo, meio ambiente e a sua história evolutiva.

Este sistema apresenta funções muito importantes para o bom estado da saúde. Permite com que o ser humano fale e cheire, aqueça o ar de acordo com a temperatura corporal, transmita oxigénio para as células, remova gases desperdiçados, protegendo também as vias aéreas de substâncias que se possam revelar tóxicas e nocivas para o organismo. [3]

Perceber a fisiologia e anatomia deste sistema acaba por se tornar um passo decisivo para um melhor diagnóstico de doenças e uma melhor perceção dos eventos que envolvem a respiração. Dominando esta área, é possível criar um sistema mais otimizado e que vá de acordo com as necessidades de cada doente que realiza tratamentos de radioterapia.

#### 2.1.1. Anatomia do Sistema Respiratório

O sistema respiratório (representado na figura 4) pode ser dividido numa zona condutora, desde o nariz até aos brônquios, que cria um caminho para direccionar os gases inalados, e numa zona respiratória (duto alveolar para os alvéolos) onde ocorrem as trocas gasosas. [4]



**Figura 4:** Anatomia do Sistema Respiratório

Os pulmões são um par de órgãos do sistema respiratório que absorvem oxigénio e eliminam dióxido de carbono, ocupando a maior parte da zona torácica. Quando ocorre o processo de inspiração, o ar



entra no corpo pelo nariz ou pela boca, desce através da laringe e traqueia, entrando nos pulmões através dos brônquios. [4]

Os brônquios principais dividem-se em unidades mais pequenas, separando-se em partes ainda menores, ou seja, os bronquíolos. Estes terminam nos alvéolos onde ocorrem as trocas gasosas, terminando todo o processo com a expiração de dióxido de carbono. [4]

### **2.1.2. Fisiologia do Sistema Respiratório**

Aliada à anatomia, temos a fisiologia deste conjunto de órgãos do ser humano. O sistema respiratório é controlado pelo sistema nervoso central (SNC). Embora exista algum controlo voluntário, este acaba por ser um processo regulado de forma automática. As partes do SNC que controlam a respiração localizam-se no tronco encefálico. Estas zonas são responsáveis pelos impulsos nervosos transmitidos em direção ao diafragma e músculos intercostais, havendo um controlo do ritmo respiratório básico. [5]

O processo do ar entrar e sair dos pulmões denomina-se de ventilação. Embora pareça um procedimento simples, fluir o ar depende de fatores como a integridade do tecido pulmonar ou a resistência ao fluxo de ar por parte das vias aéreas. [5]

Cerca de 10 a 12 vezes por minuto, num adulto, o diafragma e os músculos torácicos recebem impulsos nervosos com objetivo de haver contração. O diafragma desce e a caixa torácica sobe, aumentando o volume e criando uma pressão negativa no interior dos pulmões. Essencialmente, o ar de um ambiente de pressão elevada flui para um de baixa pressão nos pulmões. Todo este processo é conhecido como inalação, a fase ativa da ventilação. [5]

Nestes processos de inspiração e expiração, o ser humano faz circular 5 a 10 litros de ar pelos pulmões, em média. A quantidade de ar que os pulmões absorvem durante a respiração é denominada de volume corrente. Por outro lado, tem-se o volume que é recolhido ao longo de um minuto, isto é, o volume minuto. [5]

## **2.2. Sistemas de Monitorização da Respiração**

Monitorizar o sinal respiratório dos pacientes tem sido providencial no desenvolvimento da tecnologia necessária dos dispositivos, suportada pelos sistemas modernos de saúde que existem nos dias de hoje.

O subcapítulo seguinte demonstra as metodologias e sistemas de monitorização existentes para o controlo das fases de respiração de um paciente, assim como metodologias de como e quando se deve incidir radiação na zona do corpo que é necessário tratar.

### **2.2.1. Metodologias e Sistemas de Monitorização da Respiração**

Atualmente, existem diversos métodos de monitorização de respiração, mas, neste trabalho, será dado mais ênfase aos mais relevantes e com mais utilização em ambiente hospitalar.

O método ABC (*Active Breath Control*) por espirometria, ilustrado na figura 5, monitoriza e controla a respiração através da indução da apneia, de forma automática num volume de ar expirado ou inspirado e num determinado período de tempo já definidos anteriormente, havendo uma reprodutibilidade do volume pulmonar. [6]

Este género de processo pode ser utilizado em radioterapia de diferentes zonas do corpo, tal como mama ou pulmão, por exemplo. Controlando o volume do pulmão, as margens de tratamento são reduzidas, assim como, as doses de radiação em tecidos saudáveis. [6]

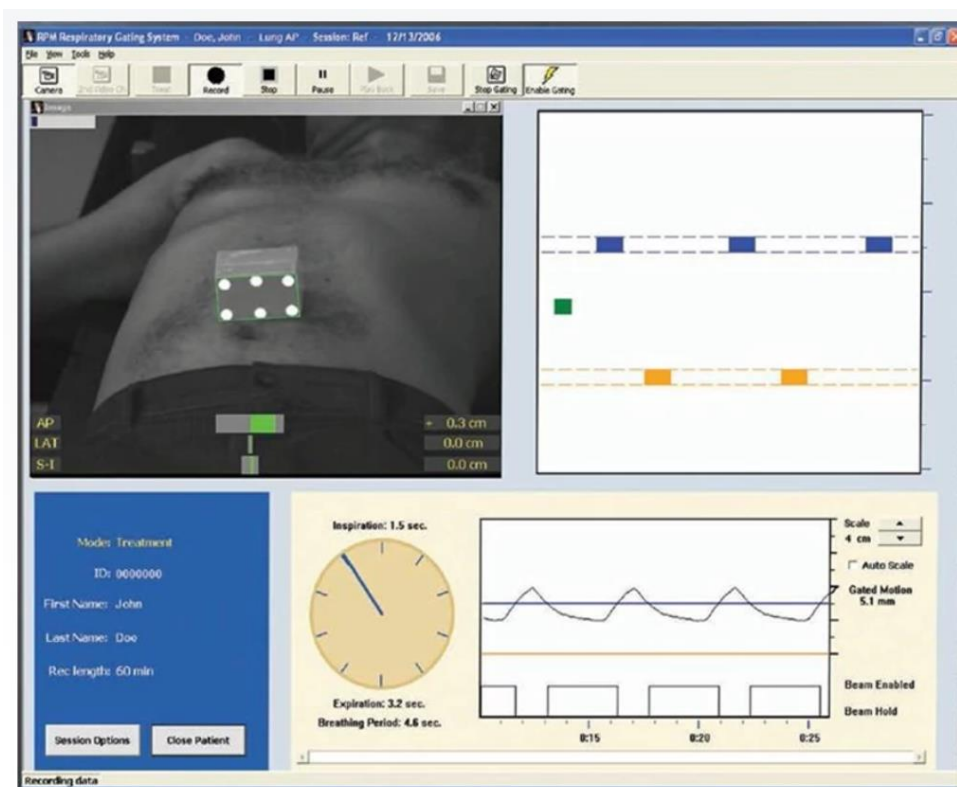


**Figura 5:** Sistema ABC

O sistema ABC é constituído por um tubo respiratório que se encontra ligado a um transdutor de volume digital, a um conjunto de captação e a uma válvula de balão, sendo que esta pode ser insuflada, com objetivo de interromper o processo de respiração. [6], [7]

O principal objetivo deste método é o surgimento de apneias reproduzíveis num respetivo volume pulmonar, seja em sessões de radioterapia ou até mesmo em exames de tomografia computadorizada (a denominada TAC). [6]

Outro método muito utilizado é com base numa câmara e um marcador. Um sistema que é constituído por esta técnica de monitorização pertence à VARIAN®, o denominado RPM (*Real-Time Position Management*). A figura 6 ilustra este exemplo de sistema.



**Figura 6:** VARIAN® RPM

O VARIAN® RPM caracteriza-se como um sistema que apresenta a possibilidade de sincronia do tratamento com a respiração. Este está aliado a uma câmara de infravermelhos (que irá incidir no bloco que se encontra no abdómen do doente) para rastrear movimentos, medindo a amplitude de respiração do paciente, com objetivo de ajudar a definir certos limites [8]. Este sistema segue os mesmos métodos de inspiração profunda mencionados no capítulo do enquadramento e pertinência.

O sistema VARIAN® RPM realiza monitorização 3D, em tempo real, da posição do paciente, acabando por detetar movimentos inesperados, o que irá revelar confiança ao aplicar o tratamento de radioterapia. [8]

Este é um sistema versátil com uma interface de imagens *On-Board Imager® kV* e *PortalVision™ MV*. Para além disso, este sistema permite associar uma verificação fluoroscópica no pré-tratamento de *gating*.

O dispositivo médico (bloco) é, normalmente, instalado sempre na mesma posição, entre o apêndice xifoide e o umbigo, que é considerada a região externa do corpo com um movimento respiratório mais significativo. Este bloco possui uns refletores de luz infravermelha. A câmara ocupa uma posição fixa em relação ao doente e está ligada a um computador que, por sua vez, se encontra interligado ao acelerador linear. [9]

O software analisa, em tempo real, a amplitude de respiração, a partir do movimento dos recetores e da respiração do doente em questão, controlando o padrão respiratório do doente e o momento em que se pode incidir radiação.

Tal como a VARIAN®, a empresa *Elekta®*, também está inserida nestes desafios da monitorização da respiração em sessões de radioterapia, apresentando técnicas similares. [9]

No caso dos sistemas desenvolvidos pela empresa *Elekta®*, no movimento respiratório é necessário que seja administrada a dose certa no alvo pretendido e, por outro lado, deve-se minimizar a dose nas zonas mais próximas consideradas críticas, como o pulmão esquerdo e o coração. Assim, os profissionais de saúde utilizam a técnica de inspiração profunda moderada (mDIBH), maximizando a dose e precisão na zona que será radiada, poupando as restantes. [10] Com esta técnica de monitorização de respiração, é possível realizar tratamentos de radioterapia apenas nos órgãos necessários, aumentando a distância para órgãos críticos que não necessitam dessa mesma radiação. [10]

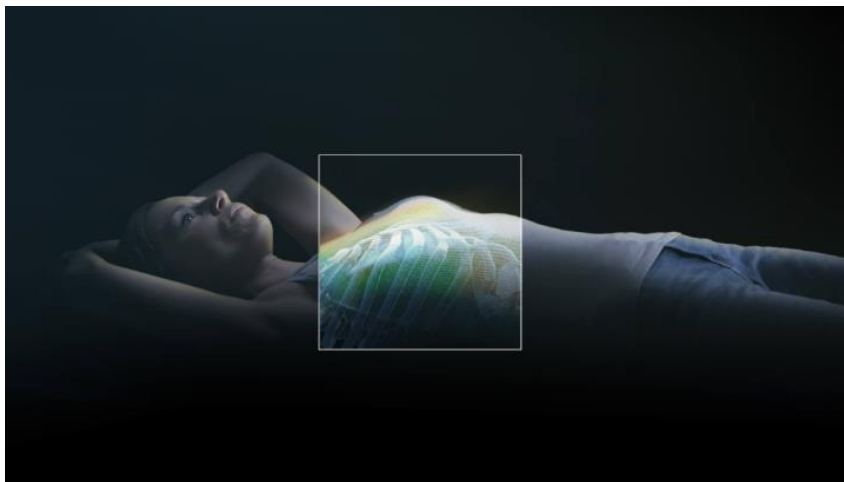
De seguida, são apresentados dois sistemas que se baseiam na utilização de raios-X para monitorizar a respiração, o *Accuray Cyberknife Synchrony®* e *ExacTrac Dynamic*.

Em relação à *Accuracy*, para detetar o movimento, há emissão de raios-X, capturando pares de imagens, sendo que o sistema kV (integrado nesta tecnologia) recolhe imagens monoscópicas sequenciais em qualquer ângulo. Este sistema apresenta um braço robótico e uma tecnologia de colimação dinâmica que permite adaptar e sincronizar a posição de onde sairá o feixe laser para a posição alvo, recentemente detetada, em tempo real, não havendo interrupção do tratamento de radioterapia. [11]

O sistema oferece precisão e exatidão aos profissionais de saúde que o utilizam, não atrasando o fluxo dos trabalhos clínicos nem prejudicando a eficiência dos mesmos. [11]

Já o sistema *ExacTrac* oferece tratamentos de alta precisão, com fluxos de trabalho variados em relação ao posicionamento e monitorização de pacientes. Este tem uma câmara de superfície térmica 4D que se articula com a utilização de raios-X utilizados em tempo real. [12]

Esta tecnologia é constituída por painéis com tamanho considerável, mostrando mais pormenor da anatomia humana, facilitando a interpretação e orientação das imagens raios-X (figura 7).



**Figura 7:** Imagem de Raios-X (*ExacTrac Dynamic*)

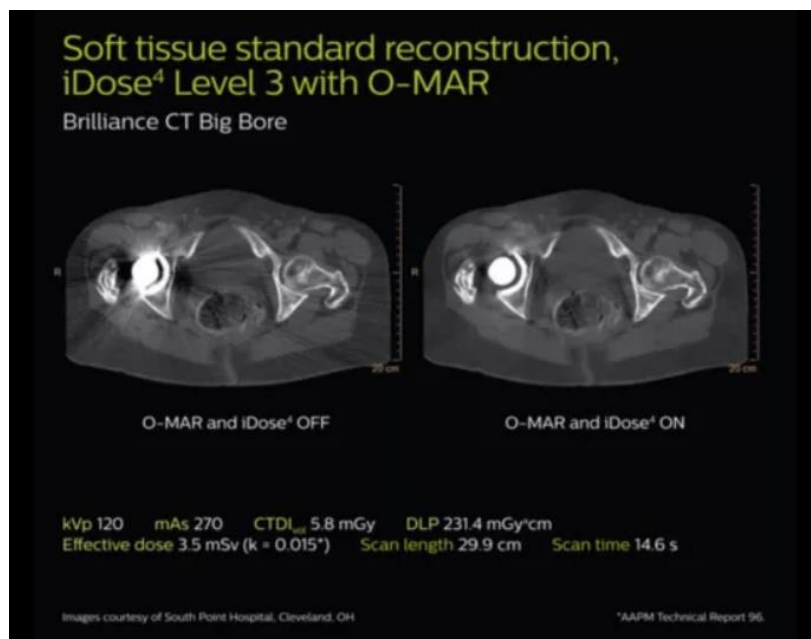
O sistema da *ExacTrac* permite, então, tratamentos de cancro de mama a partir do fluxo de trabalho de retenção da respiração por inspiração profunda e, embora seja uma técnica que envolva vários sistemas, esta tecnologia é capaz de abranger todo o posicionamento e monitorização dos respetivos pacientes. [12]

Outro sistema também utilizado atualmente é o da *Philips* que consiste num fole de ar, feito de borracha, acoplado a um transdutor de tensão. Este sistema gera um sinal que é digitalizado e transmitido ao *scanner* que efetua o registo em forma de onda. O abdómen, ao esticar e contrair, cria sinal no transdutor permitindo identificar as mudanças de pressão do ar no interior do fole. A figura 8 ilustra como os pacientes são preparados para a sessão de radioterapia, utilizando este sistema. [13]



**Figura 8:** *Philips Bellows*

O fole tem a vantagem única de ter na sua composição o *scanner Brilliance CT Big Bore* (exemplo evidenciado na figura 9). Esta tecnologia, com o seu *Big Bore*, ou seja, com maior abertura, oferece flexibilidade na execução de radiologia e em procedimentos relacionados. [13], [14]



**Figura 9:** *Brilliance CT Big Bore*

O sistema da *Philips* apresenta um campo de visão de *scan*, em tempo real, de 60 cm, encaixando-se na circunferência da maior parte dos pacientes. Para além disso, oferece precisão espacial, no posicionamento, em menos de 2 mm entre o plano que forma a imagem TC e o plano de marcação laser, suportando a precisão do posicionamento. [14]

Com a tal maior abertura, mencionada anteriormente, é possível reduzir o tempo de preparação do paciente e do próprio exame, pois existe uma maior flexibilidade no que toca à posição do paciente. [14]

Por fim, apresenta-se o método de radioterapia guiada por superfície. Na radioterapia tradicional, os doentes recebem umas mini tatuagens no local do tratamento, contribuindo para ajudar o técnico no posicionamento do doente e, posteriormente, a incidir radiação na zona correta, no entanto, neste método, tal processo não acontece. [15]

Na radioterapia guiada por superfície, são utilizadas câmaras para rastrear a superfície da pele em 3D, não havendo necessidade do paciente ficar com marcas permanentes na pele. Além disso, o método oferece vários pontos de referência na pele, havendo uma maior precisão no controlo dos movimentos, mas o doente tem a liberdade de se mover naturalmente durante o tratamento e, havendo estes pequenos movimentos, a deteção dos mesmos pode se tornar um obstáculo. Caso o doente saia da posição onde foi colocado, o sistema tem uma funcionalidade de pausa automática, contribuindo para a segurança da pessoa e a eficácia do tratamento. [15]

Os sistemas de rastreamento de superfície baseiam-se no seguinte conceito: existe uma correlação entre a superfície e o movimento interno dos órgãos, algo que não pode ser considerado como verdade absoluta, pois não se sabe o deslocamento que vai existir de cada órgão ou até do tumor, dependendo também da zona do mesmo. [16]

Os sistemas que utilizam raios-X como o da *Exactrac* são considerados uma versão diferente dos sistemas guiados por superfície, mas acabam por ser métodos mais complexos por envolverem componentes como tubos de raios-X e painéis detetores. A mesma complexidade ocorre com o sistema ABC e da *Philips*, mesmo sendo eficazes na sua tarefa de monitorização. [16]

Assim, o sistema da *VARIAN*® revela-se mais simples, com um design ao nível da interface mais intuitivo, sendo só necessário uma câmara, um marcador e um computador.

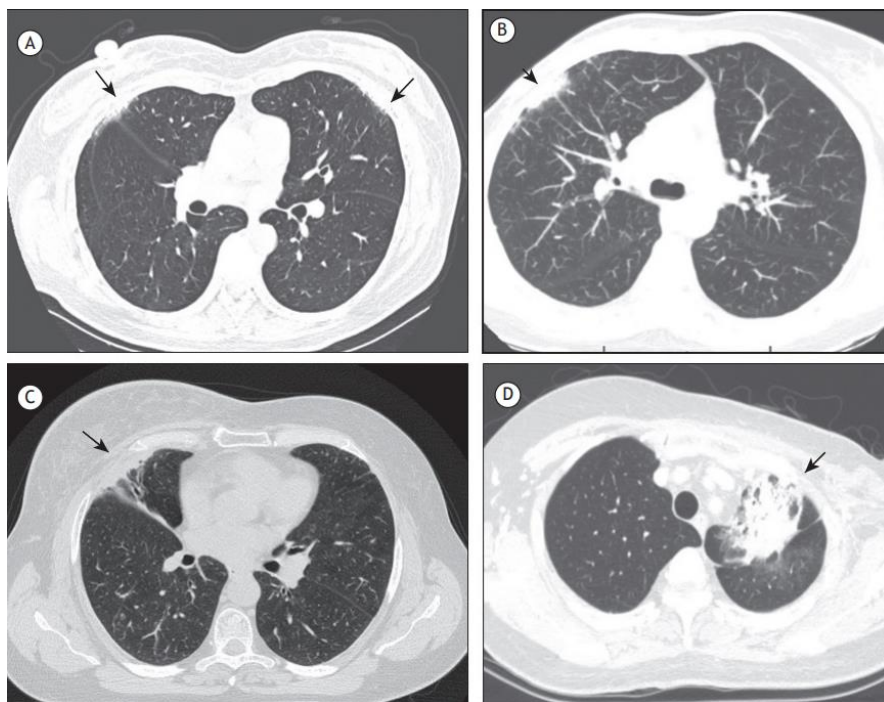
## 2.2.2. Impacto da Radioterapia na Função Respiratória

- Radioterapia Torácica

Nos tratamentos de cancro da mama, a radioterapia torácica é fortemente utilizada para reduzir riscos de recorrência locoregional e melhorar a qualidade de vida. No entanto, a radioterapia nesta zona envolve sempre riscos, mais precisamente para os pulmões e coração. [17].

A consequência mais comum é a pneumonite por radiação (PR), que, de forma geral, os pacientes começam a sentir entre um a quatro meses após o último tratamento de radioterapia. Podendo acontecer a denominada cascata de sinalização mediada por citocinas, danificando as células na zona alveolar, podendo haver uma progressão para uma inflamação respiratória aguda. O doente pode apresentar sintomas como tosse, febre baixa ou fadiga. [17], [18]

Exemplo disto está descrito num estudo cujo objetivo era verificar o impacto da radioterapia na zona torácica na função respiratória e na forma como os pacientes com cancro da mama conseguem realizar exercício físico, três meses após os tratamentos. [17] Os resultados desse estudo são revelados em imagens de TCAR (Tomografia de Alta Resolução), tal como ilustra a figura 10, realçando características de pneumonite por radiação. [17], [18]



**Figura 10:** Características de pneumonite por radiação

A imagem A foi classificada como grau 1 de pneumonite, ao contrário das imagens B, C e D que foram atribuídas um valor de grau 2, 3 e 5, respetivamente. [17]

Na primeira imagem (A), verificou-se um espessamento do septo, isto é, o tecido conjuntivo dos pulmões ficou mais espesso, resultando em opacidades reticulares. Na imagem B, notou-se um espessamento subpleural superior a 2 cm (paralelo à parede torácica). Já na imagem C, as bandas parenquimatosas do pulmão em direção à superfície pleural encontravam-se com mais de 2,5 cm (afetando o espaço alveolar). Por fim, na imagem D, está realçada uma determinada opacidade, traduzindo-se num aumento do coeficiente de atenuação dos pulmões. [17]

Durante este estudo, foram avaliados fatores como a força muscular respiratória, mobilidade torácica e foram realizados uma prova de função respiratória e um teste de exercício cardiopulmonar (antes e após a radioterapia). [17]

Os valores da força muscular e mobilidade torácica diminuíram, após 3 meses de tratamentos de radioterapia. Estes dados são explicados pela fadiga que os pacientes sentem após as sessões de tratamento, limitando as atividades diárias e atrofiando os músculos. [17]

Em relação à prova de função respiratória, houve um decréscimo significativo, após radioterapia, da CVF (indica o volume máximo de ar exalado em esforço máximo), VEF<sub>1</sub> (volume de ar expirado no primeiro segundo, de forma forçada), CPT (capacidade pulmonar total) e CI (capacidade inspiratória), com exceção da DLCO (difusão do monóxido de carbono) que diminuiu, mas não de forma tão assinalável. [17]

Já nos resultados do teste de exercício cardiopulmonar, verificou-se uma diminuição de valores como o consumo de oxigênio no pico do exercício, da ventilação/minuto e frequência respiratória. Assim, chegou-se à conclusão de que a radioterapia torácica tem influência negativa nas funções do sistema respiratório. [17]

- Radioterapia Abdominal

A radioterapia externa tem como base o posicionamento espacial preciso do tumor e tecidos. Para os doentes em que o tratamento se incide no abdômen, o movimento dos órgãos internos e o contorno externo do paciente devido à respiração, complica os processos de radioterapia para este local. [19], [20]

Em relação à radioterapia abdominal, a imobilização é um obstáculo complicado, pois o abdômen não é uma zona propriamente firme, não possui um perfil ósseo, em que um molde não irá conseguir ajustar e restringir os movimentos. Assim, a imobilização depende de dois fatores, ou seja, o conforto do paciente e a reprodutibilidade da posição do tratamento. [20], [21]

O movimento externo tem influência direta na precisão e, portanto, um estudo utilizou esta componente como forma de indicar os movimentos que ocorrem durante o tratamento, tal como avaliar a técnica de apneia na expiração, isto é, reproduzir a posição dos órgãos internos, por tratamento, na mesma fase do ciclo respiratório. O movimento dos pontos abdominais externos, anterior e lateral direitos, foi monitorizado com utilização de feixes infravermelhos. Durante o processo, foi avaliado a mudança na posição de cada ponto por respiração ou interrupção da mesma, a alteração na posição entre respirações e interrupções e a modificação na posição durante todo o período de tempo considerado. [21], [22]

O estudo mostrou duas componentes principais no movimento da posição externa em pacientes que são submetidos a tratamentos na zona abdominal. A primeira componente é o movimento regular devido à respiração e, como segundo componente, tem-se uma mudança gradual na posição externa. [22]

### 2.2.3. Movimentos Internos e Externos na Zona Torácica

A respiração segue os princípios da lei de *Boyle*, isto é, a pressão de um gás diminui com o aumento do volume, a temperatura constante. O aumento do volume da caixa torácica origina uma diminuição da pressão intratorácica, havendo influxo de ar. Por outro lado, com a diminuição do volume, devido ao relaxamento dos músculos responsáveis pela inspiração, há um aumento da pressão intratorácica, ocorrendo a expiração, sendo a parede torácica o componente essencial para todo este processo. [23]

As estruturas da parede torácica protegem órgãos como o coração, pulmões ou alguns órgãos do abdômen. Além disso, as estruturas ósseas oferecem locais de fixação para os músculos, permitindo a função mecânica do processo de ventilação. [24]

No que diz respeito à musculação da parede torácica, é possível realizar uma divisão em dois grupos, isto é, os músculos da inspiração e os responsáveis pela expiração forçada. No momento da inspiração, as zonas ântero-posterior, vertical e transversal do tórax sofrem um aumento. Quando o diafragma se contrai, um tendão (numa posição central) é tracionado inferiormente, resultando no achatamento do diafragma e no aumento das dimensões verticais do tórax. Com a contração dos músculos intercostais externos, a parte lateral das costelas eleva-se, originando um aumento do diâmetro transversal do tórax. As costelas vertebroesternais (ou verdadeiras) seguem o mesmo processo, elevando o esterno, sendo capazes de aumentar as dimensões ântero-posteriores da zona torácica. [25], [26]

Em relação à expiração forçada, diversos músculos contribuem para esse fenómeno. Por um lado, os intercostais externos avançam da costela superior para a costela inferior adjacente. Por outro lado, os músculos intercostais internos, têm a tarefa de deprimir as costelas. [27], [28]

É de realçar que pode haver momentos em que o tórax se encontra numa fase mais instável. Pode acontecer devido a fracturação de costelas em mais do que um local, originando um segmento de movimento livre. Numa fase mais inicial, o segmento é capaz de ser imobilizado pelos músculos intercostais, mas acabará por ter um movimento paradoxal em relação ao resto da parede torácica, devido à fadiga, o que irá prejudicar, de forma agressiva, o processo de ventilação. [27], [28]

### 2.3. Processamento do Sinal Respiratório

A frequência respiratória (número de respirações por minuto) é um fenómeno que exige uma regulação intensa, permitindo que as células produzam quantidades de energia suficientes, em qualquer momento que seja necessário. Um adulto em repouso, por norma, realiza 12 a 20 respirações por minuto, sendo que as crianças apresentam frequências superiores, dependendo muito das respetivas idades. [29], [30]

As respirações realizadas associam-se a um ciclo de respiração. O ciclo respiratório é constituído então pela fase de inspiração (inalação de oxigénio) e expiração (exalação de dióxido de carbono), sendo que a cada inspiração e expiração, corresponde a um ciclo de respiração. [31]

O ritmo respiratório é detetado a partir da duração, em média, do ciclo respiratório durante um determinado intervalo de tempo ou através da análise individual de cada ciclo. Já na duração individual dos ciclos, terão de ser aplicados filtros específicos, enaltecendo a frequência respiratória, a identificação de picos ou a passagem do próprio sinal pelo valor zero. [31]

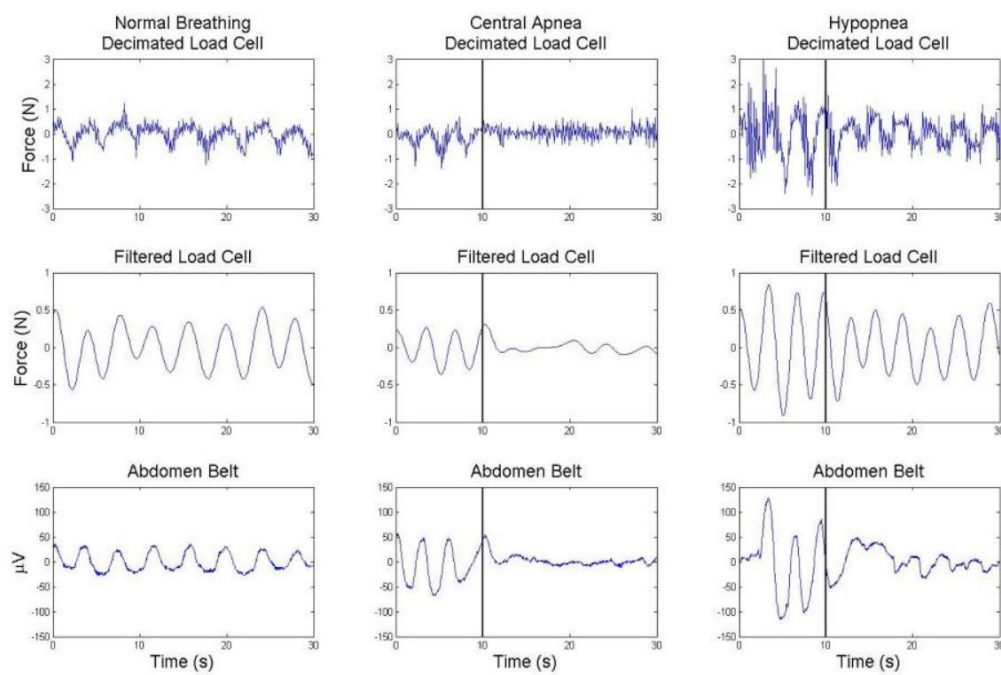
Seguidamente, são apresentados diversos estudos realizados ao longo de vários anos relativos ao processamento do sinal respiratório.

Durante o ano de 2006, utilizando células de carga inseridas numa zona mais inferior de uma cama, foi possível aceder às deformações mecânicas entre as placas de um transdutor, que ocorreram devido aos deslocamentos de massa, originados pelo processo de respiração. Os sinais recolhidos dividiram-se em corrente contínua (DC) e alternada (AC). Em relação a DC, trata-se da força aplicada ao sensor e, em AC, ocorre a variação da força tendo em conta o valor médio, que corresponde ao valor de DC. [32]

Realizou-se um sistema de coordenadas, em que nas representações dos sinais de força (no eixo dos  $yy$ ) são aplicados filtros passa-alto com uma frequência de corte igual a 0,5 Hz. Os picos são assumidos como máximos locais, sendo que a duração do ciclo respiratório é estimada através de uma contagem do número de amostras entres dois picos que ocorressem de forma consecutiva. [32]



No ano de 2009, foi publicado um artigo que mencionava a identificação de eventos respiratórios a partir de células de carga acopladas a uma cama, verificando momentos como a respiração normal, apneia obstrutiva (bloqueio das vias respiratórias durante o sono) e apneia central (paragem repentina da respiração durante o sono). [33]



**Figura 11:** Gráficos da respiração de um paciente do estudo de 2009

Na figura 11, os dados são agrupados em 3 colunas e 3 linhas. A primeira coluna refere-se à respiração normal, na segunda coluna tem-se a apneia central e, por fim, na última coluna é descrito o caso de hipopneia. Por outro lado, em relação à informação horizontal, a linha superior contém os dados decimados da célula de carga em comparação com os dados filtrados da segunda linha. Já na última linha, os gráficos referem-se a um cinto de pletismografia indutiva respiratória (colocado na zona abdominal). [33]

Ao nível do processamento do sinal, no pré-processamento, removeu-se o valor médio do sinal decimado (10 Hz), sendo filtrado por um passa-baixo *Chebyshev* tipo II de ordem 6, com uma frequência de corte igual a 0,76 Hz, atenuando o sinal em 40 dB. Em relação ao processamento, foram considerados irrelevantes os picos com uma amplitude 45% inferior em relação ao pico precedente. O sinal, de seguida, foi novamente decimado a 5 Hz (reduz-se a sua taxa/frequência de amostragem), aplicando-se o mesmo filtro, mas agora de ordem 8 (frequência de corte de 0,6 Hz). Por fim, os pontos de inflexão são considerados picos quando a amplitude é superior à tendência e vales em caso contrário, sendo que a taxa de respiração é então calculada pela duração média entre estes dois termos mencionados. [33]

Em 2014, foi apresentado um sistema para avaliar a atividade respiratória a partir dos sinais recebidos de sensores ligados a uma cama, mas com aplicação de diversos filtros passa-baixo, com frequências de corte diferentes. No pré-processamento, avaliou-se o sinal em janelas, removendo-se as zonas em que os valores pico-a-pico eram superiores ao dobro do valor da média. Numa fase posterior, filtrou-se o sinal através de diversos passa-baixos com frequências de cortes equivalentes a 0,154Hz, 0,22Hz, 0,33Hz e 0,5Hz. [34]

Identificou-se que o fenómeno da respiração ocorre entre os valores de frequência de 0,1 e 0,5 Hz e que as distorções se sucedem em frequências 1,5 vezes superiores ao valor natural, motivando assim a escolha desses valores de frequência. Os picos/vales foram localizados a partir dos mínimos e máximos

locais, sendo que a duração do ciclo respiratório calculou-se através do intervalo entre dois picos consecutivos. Por outro lado, a amplitude foi medida pela diferença entre um vale e um pico. Com este estudo, assumiu-se que o ritmo respiratório considerado mais “correto” correspondia ao sinal com menor variação na amplitude, tendo em consideração os 5 ciclos respiratórios precedentes ao respetivo intervalo. [34]

Também foi desenvolvido um sistema, no ano de 2015, para aceder aos parâmetros respiratórios com a união entre os sinais de um eletrocardiograma e sistemas com base no efeito cardiobalístico (ocorre com as contrações do ventrículo esquerdo). Neste caso, o sistema apresentado é constituído por 2 sensores e 8 filtros passa-banda, sendo que os parâmetros respiratórios são identificados com a utilização de *clusters*. Numa fase prematura, os dois sinais são filtrados por 4 passa-bandas e um *Butterworth* de ordem 4 com diferentes intervalos de frequências de corte como 0,05 e 0,154 Hz ou 0,1 e 0,33 Hz. [35]

O objetivo da utilização dos filtros passa-banda passou por evitar a ocorrência de distorções na amplitude dos sinais por consequência das atenuações nas bandas de rejeição dos filtros passa-baixo (utilizados por norma). Seguidamente, extraiu-se o máximo e mínimo e as extremidades de subida e descida. Com estes dados, o algoritmo sugeriu intervalos de repetição para cada dado retirado, tendo sido obtido 32 sugestões (guardadas, em forma de vetores, nos referidos *clusters*) de duração do respetivo ciclo respiratório. O ritmo respiratório é ponderado a partir das 6 sugestões mais próximas, umas em relação às outras, desvalorizando os valores que desviam por mais de 1 segundo em relação aos valores periféricos. [35]

Por fim, no ano de 2016, novamente com a utilização de células de carga, identificou-se um sensor para analisar o ritmo respiratório. Numa fase inicial, os sinais são filtrados por passa-baixos *Butterworth* de ordem 5, apresentando uma frequência de corte equivalente a 0,5 Hz. Estes sinais foram submetidos a uma decomposição de modo empírico, com objetivo de retirar tendências residuais. De seguida, ocorreu a identificação da periodicidade dos sinais nas frequências expectáveis, isto é, entre 0,2 e 0,8 Hz, calculando-se indicadores de qualidade através da autocorrelação e densidade espectral de potência (funções matemáticas para análise de sinais). [36]

Neste trabalho, foi considerado um valor padrão de 1,25, isto é, todos os sinais que apresentassem indicadores de qualidade (autocorrelação + densidade) inferiores ao valor expectável, eram considerados como ruído, com pouca utilidade. Com a identificação dos picos dos sinais, foi possível ponderar o ritmo respiratório. [36]

Nos estudos referidos, utilizaram-se filtros passa-alto, passa-baixo (como o *Butterworth*), passa-banda e um passa-baixo *Chebyshev*. Um filtro passa-alto é capaz de permitir a passagem de altas frequências (superiores a uma determinada frequência de corte), atenuando as baixas frequências (inferiores à frequência de corte), surgindo como seu complemento o filtro passa-baixo que realiza o processo inverso. [37]

Já o filtro passa-banda apenas deixa passar determinadas frequências, rejeitando as restantes. Por fim, tem-se o filtro passa-baixo *Chebyshev* que é utilizado para diferenciar frequências de bandas diferentes e o filtro passa-baixo *Butterworth* que oferece um ganho constante de frequências numa banda específica. [37]

Quer com a utilização de diferentes filtros, diferentes frequências, sensores, células de carga e até com contribuição da monitorização de um eletrocardiograma, cada estudo obteve o seu método mais

eficaz para conseguir identificar o ritmo respiratório, tendo em conta a fase de evolução da medicina e recursos disponíveis, em cada ano respetivo.

Dado que cada estudo conseguiu, com cada técnica individual de processamento de sinal, analisar o ritmo respiratório, os filtros mencionados são considerados um bom ponto de partida para aplicar em sistemas de monitorização da respiração.

### 3. MÉTODOS E APLICAÇÃO

Inicialmente, foram definidos determinados objetivos para o software funcionar de acordo com as necessidades específicas deste tipo de tratamento. Era necessário haver uma boa detecção do objeto-alvo para utilizar aquando da monitorização do doente, uma boa calibração do sistema e criar uma interface gráfica para o utilizador e doente. Todas as ferramentas incluídas na interface foram criadas de acordo com o que é requerido para a técnica utilizada, tendo o objetivo de melhorar o tratamento do próprio doente, assim como ser possível o controlo e monitorização por parte da equipa técnica.

Atualmente, o IPO-Porto utiliza o sistema de monitorização da VARIAN® RPM (mencionado na secção da revisão bibliográfica) em apenas dois dos seus oito equipamentos em radioterapia, em que é colocado um objeto no abdómen do paciente e feita monitorização através de um sistema vídeo.

O RPM® é um sistema simples de usar e fácil de implementar, pois, são apenas necessários uma câmara e um bloco com marcadores, interligados a um software. Tem como desvantagem o custo e a fragilidade dos marcadores refletores de infravermelho, que são facilmente afetados pelo contacto com as mãos, correndo-se o risco de passarem a não ser detetados pela câmara por não refletir o feixe infravermelho, comprometendo a detecção do sinal. Por fim, o facto de o sistema identificar apenas estes marcadores, faz com que a aquisição da imagem apenas possa conter o marcador e não qualquer outro ponto brilhante, caso contrário não é identificado.

Então, decidiu-se optar por tentar desenvolver e implementar outro método de detecção do sinal respiratório. Algo que se revelasse simples, de acesso gratuito e que, posteriormente, o IPO-Porto pudesse implementar e, claro, partilhar essa informação, dando o seu feedback sobre o tema e proporcionando a outras entidades o poderem realizar sistemas com ideias semelhantes.

Assim, foi colocada a hipótese de implementar um sistema similar nos equipamentos desprovidos de monitorização, adicionando a possibilidade da passagem de informação para o doente. Além disso, conseguiriam um sistema pessoal e não dependeriam de terceiros para realizar as suas sessões de radioterapia externa, tornando-se também menos dispendioso.

Para cumprir com os objetivos foi decidido realizar um sistema de monitorização num ambiente de linguagem de programação em *Python*, visto que é uma linguagem gratuita, de fácil aprendizagem, constituída por várias bibliotecas e funcionalidades, podendo ser usada em diferentes sistemas operacionais. [38]

Tendo em conta o que foi proposto e, para criar um sistema desta dimensão, chegou-se à conclusão que o equipamento utilizado deveria cumprir todos os requerimentos de segurança e aplicabilidade, tendo que passar por um processo de calibração para se assegurar a fiabilidade dos resultados medidos e seria necessária uma boa detecção do comportamento respiratório, juntando tudo numa interface intuitiva e fácil de utilizar.

Aqui nasce a ideia do rastreamento de padrões físicos, padrões esses que existem em bibliotecas de linguagem *Python* e a informação que contém pode ser calibrada e detetada através de uma câmara.

#### 3.1. Marcadores Físicos

Como método principal para este projeto teve-se em consideração os fundamentos da realidade aumentada, que proporciona informação visual, através do computador, oferecendo conhecimento sobre o que está a ocorrer no mundo real. No entanto, um dos principais desafios desta área é estimar, em tempo real, a posição e orientação da câmara que está a ser utilizada para o sistema em específico. [39]

A solução encontrada para esse problema foi o rastreamento de marcadores. Estes sistemas apresentam, como base de funcionamento, o rastreio de padrões físicos, com objetivo de identificar objetos e estabelecer a relação existente entre o que é virtual e real. Os padrões encontram-se armazenados em bases de dados e após o seu reconhecimento, é possível estabelecer uma ligação entre o mesmo e a respetiva câmara. Associadas às bases de dados estão as denominadas bibliotecas, que a partir de imagens, calculam a posição e orientação do marcador em relação à câmara. [39]

Assim, a partir da linguagem de programação *Python*, considerou-se a utilização destes marcadores físicos, tendo sido escolhido o denominado marcador *ArUco*.

Um marcador *ArUco* é considerado um padrão que apresenta como seu conteúdo, uma borda preta e uma matriz binária (parte branca) que, de acordo com a forma apresentada, é-lhe atribuído uma identificação. A parte preta facilita uma deteção mais rápida e, por outro lado, o código binário contribui para descobrir erros e corrigir os mesmos. Este marcador pode apresentar vários tamanhos, como por exemplo 4x4, tendo como significado um tamanho composto por 16 bits.

Nestes padrões, estão incluídos os denominados dicionários predefinidos que constituem tamanhos variados de marcadores. Para o projeto, foi utilizado um marcador *ArUco* 6x6\_250, ou seja, um dicionário constituído por 250 marcadores e um tamanho de 6x6 bits, estando identificado com o número 1. A figura 12 ilustra o padrão *ArUco* utilizado para este trabalho.

Além do que foi referido anteriormente, a imagem pode ter qualquer informação que não limita a aquisição e seguimento deste padrão em específico.

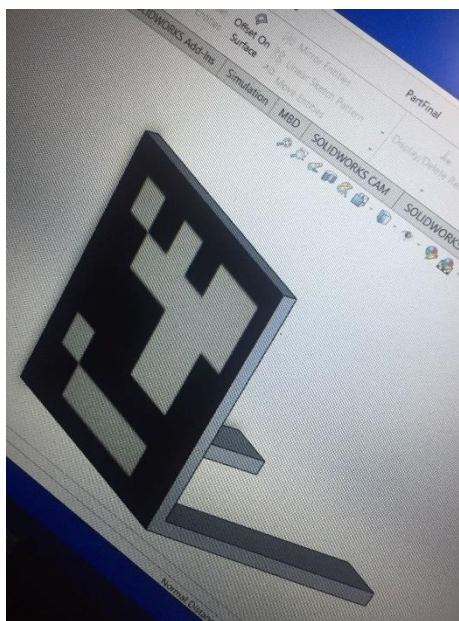


**Figura 12:** Marcador *ArUco*

Assim, com a programação em *Python*, foi possível utilizar este marcador e haver um reconhecimento do seu conteúdo através de uma câmara colocada na sala de tratamento, dando início à deteção e representação do padrão respiratório de cada paciente.

A partir deste marcador, decidiu-se testar a criação de um objeto 3D, para colocação na superfície do doente, com o desenho do marcador já implementado na face frontal. As dimensões e a forma do mesmo seriam semelhantes ao que já existe, não influenciando a passagem de radiação.

Posteriormente, com o auxílio do software *SolidWorks*, desenhou-se um objeto em forma de L, tal como mostra a figura 13.

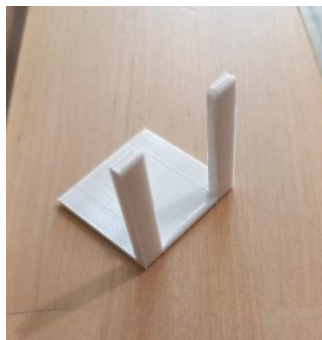


**Figura 13:** Desenho no *SolidWorks*

O formato escolhido teve em consideração o contacto que o objeto tem com o abdómen do doente. Há uma menor área de contacto com o corpo do paciente, não criando zonas radioopacas sem comprometer a segurança de colocação do objeto na zona indicada. No caso da figura 13, o desenho na

face é meramente ilustrativo, pois o objetivo seria incorporar na própria face os quadrados brancos e pretos correspondentes ao desenho deste marcador.

De seguida, transferiu-se o ficheiro do *SolidWorks* para um software de uma impressora 3D e obteve-se o seguinte resultado, ilustrado na figura 14.



**Figura 14:** Objeto impresso

O objeto foi criado, faltando a implementação do próprio padrão com a impressão a duas cores diferentes. Como consequência, os testes foram realizados através de um padrão 2D colocado na face frontal de um marcador.

### 3.2. Interface Gráfica

A interface gráfica é constituída por quatro componentes, nomeadamente a visualização em tempo real do marcador, um gráfico que representa a curva da respiração do doente, uma imagem que indica a ausência ou não de radiação e, por fim, um conjunto de botões representativos da base de dados inserida neste projeto.

O código principal é uma aplicação em *Python* usando a biblioteca *Tkinter* para criar uma interface gráfica e, a partir da biblioteca *OpenCV*, capturar vídeo da webcam e realizar a deteção de um marcador *ArUco*, sendo este utilizado para monitorizar o padrão de respiração de um paciente. O sistema exibe um gráfico em tempo real que representa a amplitude da respiração ao longo do tempo.

Como qualquer código de uma aplicação, são utilizadas funções de maneira a ter uma melhor organização, sendo cada função responsável por diferentes tarefas. Em destaque, estão duas funções principais do programa, que é a que tem capacidade em detetar o marcador *ArUco* e outra que atualiza o gráfico da respiração.

Em relação à deteção do marcador, há uma captura da imagem através da câmara disponível, de seguida converte-se para uma escala de cinza, deteta-se o padrão, rastreando o mesmo. A deteção baseia-se na localização dos quatro cantos do marcador *ArUco*, desenhando linhas que vão unir esses cantos, formando um quadrado à volta do objeto, baseando-se em coordenadas que se referem ao centro da imagem.

Quanto à função que realiza a atualização do gráfico, à medida que o marcador sobe ou desce na imagem, é desenhada uma curva que segue esses mesmos movimentos, ilustrando o padrão de respiração do doente em questão. Esta função também permite que o utilizador ajuste determinados limites superior e inferior, além da linha de base, utilizando os botões do rato. Os limites superior e inferior vão delimitar a zona de radiação do paciente, ou seja, só quando a curva da respiração atinge essa zona é que é possível incidir radiação no doente. A linha de base funciona como uma linha ou um valor de referência da curva, podendo também ser ajustada, de acordo com o comportamento da curva em amplitudes mais reduzidas.

Ainda nesta função, também é atualizada a imagem da radiação quando a curva está dentro ou fora dos limites mencionados anteriormente.

Associadas a estas funções principais, existe uma que filtra o sinal, tornando-o mais suave, evitando o máximo de ruído possível. Numa fase inicial, aplicaram-se alguns filtros (como um passa-baixo ou passa-alto) mencionados no capítulo da revisão bibliográfica, mas resultaram em problemas ao nível do ruído, cálculo de valores e um movimento atrasado da curva em relação ao marcador. Assim, aplicou-se o denominado filtro EWMA (*Exponentially-Weighted Moving Average*) que suaviza um ponto pertencente aos dados disponíveis, calculando a média, de forma exponencial, desse ponto específico em conjunto com todas as medições anteriores, acabando por resolver os problemas referidos. Além disto, consome uma memória muito reduzida e tem um método de trabalho mais rápido, apresentando um fator de suavização de sinal que pode ser modificado no código. [40]

Por fim, associado à interface, criou-se uma base de dados que terá no seu conteúdo variáveis como o nome do paciente, o número associado e os seus limites superior e inferior. O nome e número do doente são inseridos manualmente pelo utilizador, mas os limites são guardados depois de se realizar a sessão de referência (sessão primária realizada pelo paciente, onde são avaliados os parâmetros do seu comportamento respiratório e posteriormente utilizados nas sessões de tratamento seguintes).

### 3.3. Calibração

Ter noção da informação espacial e obter um sistema de coordenadas, ou seja, se o objeto se move, por exemplo, 2 cm no eixo positivo dos  $yy$ , revela-se uma tarefa muito importante para um sistema deste género. Quando se monitoriza a respiração do doente, o objeto irá realizar diversos movimentos e os mesmos devem ser quantificados, contribuindo para um resultado mais fidedigno.

Assim, é necessário que haja uma correlação entre os movimentos ocorridos e os respetivos deslocamentos e é a partir desse pressuposto, que surge a necessidade de realizar calibração.

Em relação ao código principal, estão associados três ficheiros responsáveis pela calibração do marcador *ArUco*.

Primeiramente, são recolhidas imagens em tempo real a partir da deteção de um padrão quadriculado (figura 15), sendo que essas imagens são guardadas numa pasta e utilizadas posteriormente. Numa segunda fase da calibração, é lida a informação que as imagens contêm, os cantos do padrão são identificados e a calibração é realizada com base nesses pontos.

Quer o marcador *ArUco*, quer o padrão quadriculado, apresentam dimensões específicas, cujos valores são introduzidos no código, com objetivo de haver uma calibração o mais correta possível.

Dados como matriz de calibração, coeficientes de distorção e vetores de translação e rotação são guardados igualmente. Por fim, é possível estimar a posição do marcador, utilizando todas as informações anteriormente mencionadas. No aspeto visual, é fornecida a identificação do marcador, a distância em relação à câmara e a sua posição em  $x$  e  $y$ , em tempo real (figura 17).



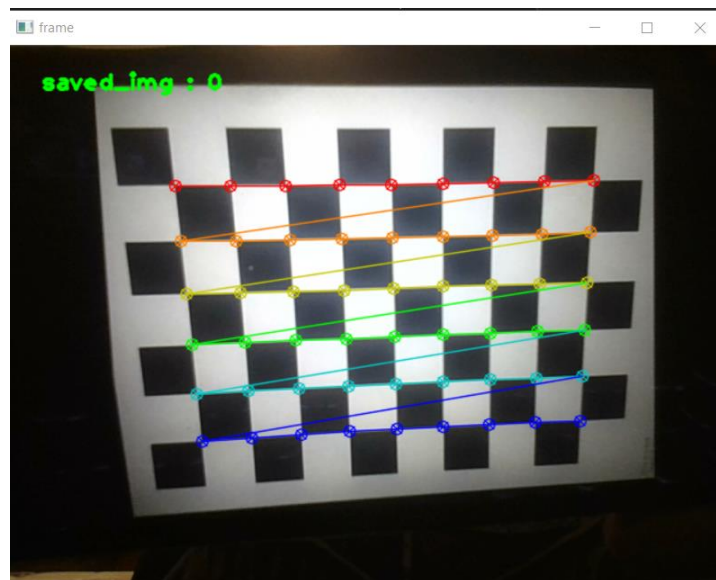
## 4. RESULTADOS E DISCUSSÃO

Com base nos conceitos referidos no capítulo anterior, desenvolveu-se um software com todas as funcionalidades necessárias e requisitadas para se monitorizar a respiração de cada doente que faça sessões de radioterapia, de uma forma simples e eficaz. De seguida, são apresentados os resultados obtidos, sendo que no final de cada capítulo de resultados é apresentada uma discussão sobre os mesmos.

### 4.1. Janela de Calibração

Inicialmente, é necessário realizar a calibração do marcador *ArUco* antes de realizar uma monitorização do sinal respiratório. A calibração, como foi dita anteriormente, apresenta 3 fases, ou seja, a fase de recolha de imagens, da leitura da informação fornecida pelas imagens e, com base nestas primeiras duas fases, existe uma terceira onde é apresentada uma informação visual sobre as coordenadas do marcador e a distância do mesmo à câmara.

A figura 15 ilustra a primeira fase do processo de calibração associado ao software.



**Figura 15:** Fase 1 da Calibração

Como se verifica pela imagem, são detetados os cantos do padrão quadriculado num *loop* infinito, estando associado a um contador de imagens no canto superior esquerdo. O objetivo desta fase é guardar o maior número de imagens possível e em posições diferentes, para que a calibração seja o mais precisa

e correta. O utilizador recolhe a imagem numa determinada posição e vai variando a mesma, obtendo mais do que uma perspetiva do padrão.

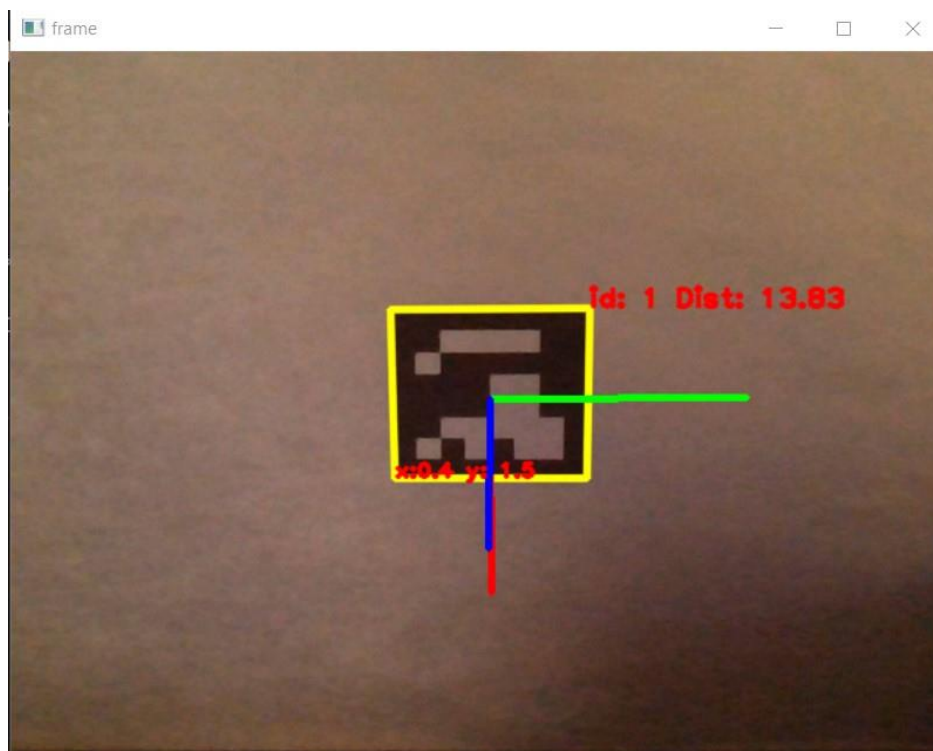
Na segunda fase, com base nas imagens recolhidas, são obtidos os dados no terminal do *Visual Studio Code* (programa utilizado para elaborar o código de todo o software). A informação fornecida é apresentada de seguida na figura 16.

```
[176. 88. 0.]
[ 0. 110. 0.]
[ 22. 110. 0.]
[ 44. 110. 0.]
[ 66. 110. 0.]
[ 88. 110. 0.]
[110. 110. 0.]
[132. 110. 0.]
[154. 110. 0.]
[176. 110. 0.]
image0.png
image1.png
image10.png
image11.png
image12.png
image13.png calibrated
image14.png dumping the data into one files using numpy
image15.png -----
image16.png loading data stored using numpy savez function
image17.png
image18.png
image19.png
image2.png loaded calibration data successfully
```

**Figura 16:** Dados da Fase 2 da Calibração

Assim, são obtidos os valores da matriz de calibração, coeficientes de distorção e vetores de rotação e translação da câmara, associados a cada imagem que foi guardada no processo anterior, sendo que, a partir deste momento, a câmara está calibrada e os dados de calibração foram guardados com sucesso.

Na fase final da calibração, é iniciada nova captura de vídeo e, tendo como base os dados de calibração, o marcador é detetado em cada *frame*, sendo estimada a posição e orientação do mesmo em relação à câmara (figura 17).



**Figura 17:** Fase 3 da Calibração

Se o marcador for detetado, é estimada a posição e orientação, retornando os vetores de translação e rotação. Os cantos são desenhados, assim como os eixos das coordenadas associados ao respetivo *frame*. Numa fase posterior, os movimentos face à câmara são calculados com base nos vetores de translação.

No caso da figura 17, o marcador *ArUco* é identificado com o número 1, com umas coordenadas de 0,4 cm no eixo dos  $xx$  e 1,5 cm no eixo dos  $yy$ , encontrando-se a 13,83 cm da câmara.

Em termos práticos, foram realizados vários testes para testar a veracidade dos valores da calibração ilustrados na figura 17.

A experiência realizada para testar a calibração consistiu em colocar o marcador em 4 posições diferentes (nos cantos de placas quadradas com 1 cm de altura e 30 cm de lado). De seguida, colocaram-se 10 placas empilhadas, estando o marcador situado nas 4 posições e retirava-se as suas coordenadas para cada uma. Após o registo dos dados da quarta posição, retirava-se uma placa, baixando a altura a que o bloco se encontrava e assim sucessivamente. Um exemplo do que foi realizado está ilustrado na figura 18. A posição 1 é que a está ilustrada na figura, sendo que a segunda posição é o outro canto 30

cm para a esquerda, ou seja, percorrendo os cantos das placas no sentido contrário dos ponteiros do relógio.



**Figura 18:** Teste da Calibração

Os valores de  $x$  e  $y$  recolhidos encontram-se registados nas tabelas 1 e 2, sendo que o número de experiências são representadas por “E” e as quatro posições por “P”.

**Tabela 1:** Coordenadas  $x$  do marcador

$x/cm$	P1	P2	P3	P4
E1	7,2	-20,8	-18,8	8
E2	7,1	-20,3	-19	8,1
E3	7,1	-20,6	-19,2	8,4
E4	7,1	-20,1	-19,1	8,2
E5	7,3	-20,7	-18,7	8,2
E6	7,2	-19,7	-18,8	8,1
E7	7,2	-19,7	-19,2	8,2
E8	6,9	-20,8	-19,2	8,2
E9	7,3	-20,4	-19,2	8,2
E10	7,1	-20,7	-19,1	8,4
<b>Média</b>	7,15	20,38	19,03	8,2

**Tabela 2:** Coordenadas  $y$  do marcador

$y/cm$	P1	P2	P3	P4
E1	0,5	0,6	0,5	0,5
E2	-0,6	-0,6	-0,7	-0,7
E3	-1,6	-1,7	-1,8	-1,8
E4	-2,6	-2,6	-2,7	-2,8
E5	-3,7	-3,8	-3,8	-3,8
E6	-4,8	-4,8	-4,9	-4,9
E7	-5,9	-5,9	-6	-5,9
E8	-6,9	-7	-6,9	-7
E9	-8,1	-8	-8,1	-8,1
E10	-9,1	-9,2	-9,1	-9,2

Com base nestes dados, realizou-se mais duas tabelas. A tabela 3, que mostra a diferença entre os valores de  $x$ , entre as diferentes posições (cada placa com 30 cm e o bloco com 3 cm de lado) e a tabela 4, onde é evidenciada a diferença das coordenadas no eixo dos  $yy$ , entre as respectivas alturas.

**Tabela 3:** Análise de erros na coordenada  $x$ 

$\Delta x/cm$	P2-P1	P4-P3	Diferença Teórica	E2-1 / %	E4-3 / %
E1	28	26,8	27	3,70	0,74
E2	27,4	27,1	27	1,48	0,37
E3	27,7	27,6	27	2,59	2,22
E4	27,2	27,3	27	0,74	1,11
E5	28	26,9	27	3,70	0,37
E6	26,9	26,9	27	0,37	0,37
E7	26,9	27,4	27	0,37	1,48
E8	27,7	27,4	27	2,59	1,48
E9	27,7	27,4	27	2,59	1,48
E10	27,8	27,5	27	2,96	1,85
<b>Média</b>				2,11	1,15

**Tabela 4:** Diferenças de coordenadas do  $y$ 

$\Delta y/cm$	E2-E1	E3-E2	E4-E3	E5-E4	E6-E5	E7-E6	E8-E7	E9-E8	E10-E9
P1	1,1	1	1	1,1	1,1	1,1	1	1,2	1
P2	1,2	1,1	0,9	1,2	1	1,1	1,1	1	1,2
P3	1,2	1,1	0,9	1,1	1,1	1,1	0,9	1,2	1
P4	1,2	1,1	1	1,1	1,1	1	1,1	1,1	1,1

Analisando a tabela 1, verificou-se que os valores  $x$  nas diferentes posições e alturas encontram-se relativamente perto da respetiva média, o que faz com que os desvios médios não sejam muito significativos.

Ao mudar o marcador da posição 1 para a 2 e da posição 3 para a 4, o mesmo desloca-se na teoria 27 cm (30 cm da placa – 3 cm do bloco) para a esquerda e, posteriormente, para a direita. Na tabela 3, verificou-se a diferença entre os valores de  $x$  nessas posições e calculou-se uma diferença para averiguar se se deslocava ou não os 27 cm. Entre a primeira e segunda posições, a diferença oscilou entre os 27 e 28 cm, correspondendo a um erro relativo de 2,11% (aproximado às centésimas). Por outro lado, entre as posições 3 e 4, os valores rondam os 26 e 27 cm, sendo um erro aproximadamente de 1,15%.

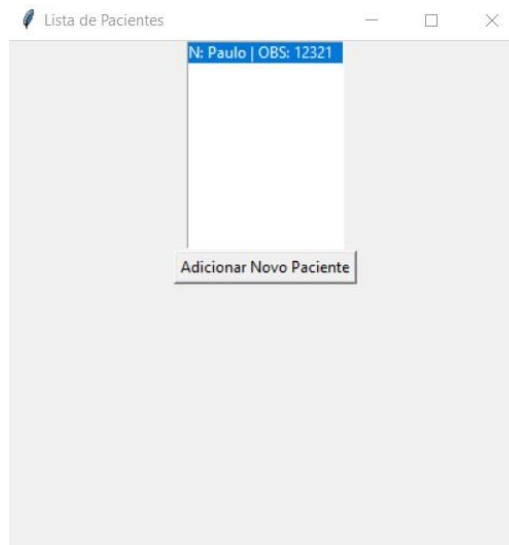
Os valores dos erros, apesar de serem reduzidos, podem ser justificados por alguma falta de estabilidade das placas todas juntas com o bloco por cima, daí, na imagem, aparecerem outros objetos naquela zona, com objetivo de criar mais peso e não haver tantas oscilações. Para além disso, a posição da câmara acaba também por influenciar os resultados, assim como seu alcance e resolução.

O valor mais relevante que se queria analisar era o valor do marcador no eixo dos  $yy$  (tabela 2), pois é o movimento vertical mais relevante para o trabalho em questão. O marcador será colocado no abdómen do paciente e, aquando dos movimentos de inspiração e expiração, o marcador realizará os tais movimentos na vertical. Neste caso, calculou-se a diferença entre os valores de  $y$  (tabela 4), entre duas respetivas alturas, tendo como base o valor teórico de 1 cm (altura da placa). Estes revelaram ser resultados satisfatórios pois todos os valores encontram-se muito perto desse valor teórico (um valor mínimo de 0,9 e um máximo de 1,2), sendo que alguns deles correspondem precisamente ao mesmo. A diferença de 0,1 ou 0,2 cm acaba por ser justificada pelas mesmas razões referidas anteriormente em relação aos valores de  $x$ .

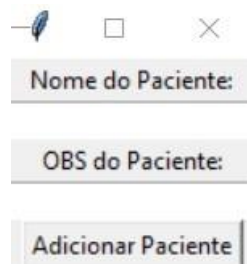
## 4.2. Software Principal

Com a calibração feita, é possível avançar para o software em si. Este é composto essencialmente por duas janelas, uma dedicada à lista de pacientes existentes na base de dados e outra que pode ser considerada a janela principal, onde se realiza a monitorização da respiração do doente.

As figuras 19 e 20 referem-se à parte responsável por apresentar e adicionar doentes à base de dados.



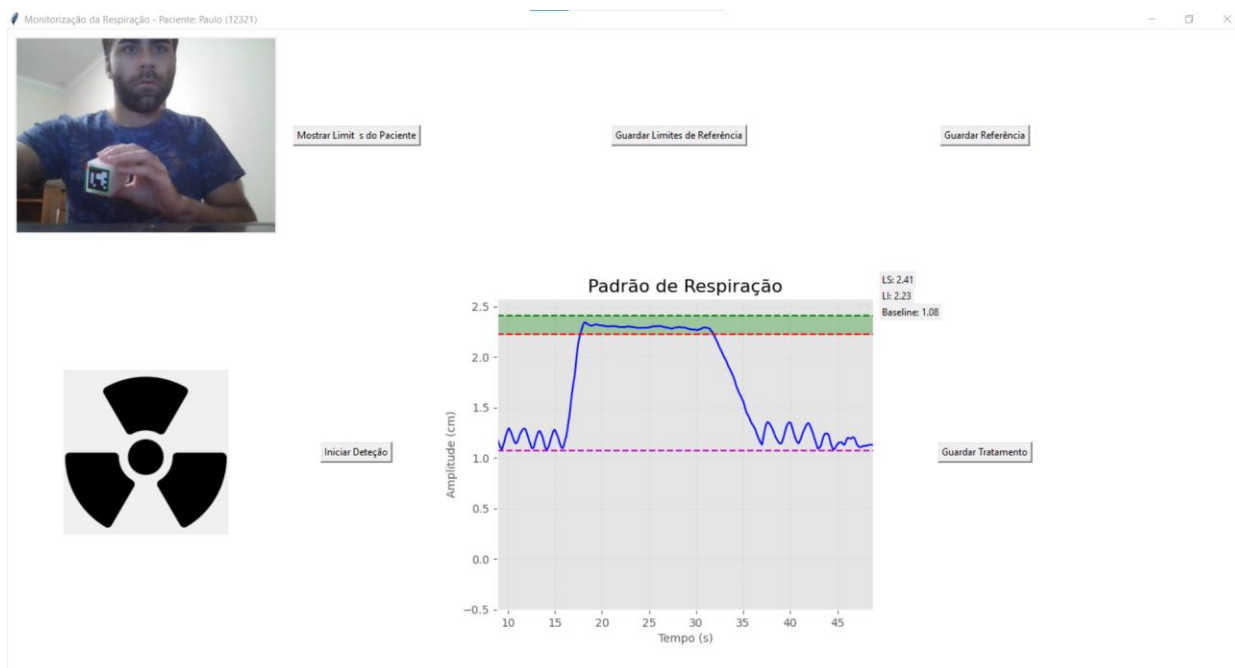
**Figura 19:** Janela da Lista dos Pacientes



**Figura 20:** Janela para Adicionar Novo Paciente

Dentro do retângulo branco são apresentados os nomes e respetivo OBS (número de identificação do doente) que se encontram registados na base de dados. Ao clicar no botão “Adicionar Novo Paciente”, esses mesmos campos são preenchidos, para depois aparecer um novo dado no dado retângulo.

Na figura 19, é possível verificar que o paciente existente está selecionado. Ao ser selecionado, na janela principal de monitorização (figura 21), no cabeçalho irá aparecer as informações do doente selecionado, ou seja, nome e número de identificação. Assim, existe sempre informação visual de qual é a pessoa que está a ser tratada naquele preciso momento.



**Figura 21:** Janela de Monitorização da Respiração

Como se pode verificar pela figura 21, o software é constituído essencialmente por três componentes, isto é, câmara, símbolo de radiação e gráfico, sendo que estão ligados a cinco botões.

O botão “Iniciar deteção” é responsável por iniciar o reconhecimento do marcador, e por consequência, o desenho da curva. Por sua vez, os botões “Guardar Referência” e “Guardar Tratamento” guardam os dados do gráfico num ficheiro de texto, em relação a sessões de referência e de tratamento do doente, respetivamente. No fim da sessão de referência, é possível guardar na base de dados os limites superior e inferior do doente com o botão “Guardar Limites de Referência”, com objetivo de poderem ser consultados nas sessões seguintes de tratamento, clicando no botão “Mostrar Limites do Paciente”.

A aplicação desenvolvida apresenta um design simples e intuitivo com todas as funcionalidades que um sistema de monitorização da respiração poderia necessitar. A curva do gráfico segue os movimentos corretos do marcador e em tempo real, sendo representado um sinal com o mínimo ruído possível, com um comportamento de uma onda sinusoidal. É possível observar que a primeira fase da curva corresponde à respiração livre do doente.

Depois, é colocada uma *baseline* (linha roxa), tendo em conta os primeiros mínimos representados naquela sequência, isto porque, o doente, por norma, na sua respiração livre, pode apresentar máximos diferentes, mas os mínimos acabam por ser consecutivos e então são os ideais para traçar uma linha de base sobre esses valores, sendo o nosso valor de referência da sessão.

De seguida, o doente entra na fase de inspiração profunda, sendo que a sua curva de respiração sobe e acaba por atingir um determinado valor, até que se mantém praticamente constante durante uns segundos (momento da apneia, em que sustém a respiração), acabando a curva por descer, estando o doente na fase de expiração e volta a repetir o mesmo processo, iniciando novamente na respiração livre.

Tendo em conta o comportamento do doente na inspiração profunda, são colocadas duas linhas que representam o limite superior (linha verde) e o limite inferior (linha vermelha), sendo considerado o intervalo de respiração profunda do paciente. Essencialmente, o doente quando voltar a entrar em inspiração profunda, a sua curva de respiração deve manter-se entre aqueles valores, para que possa ser



incidida radiação no momento certo (a imagem do lado esquerdo do gráfico ativa-se, mudando de cor, quando o doente está dentro dos limites, indicando ao técnico que o paciente pode receber radiação).

Assim, o programa é capaz de controlar e monitorizar os momentos de respiração do doente, sendo benéfico tanto para o mesmo, como para os técnicos que têm ao seu dispor um software fácil e simples de utilizar.

## 5. CONCLUSÃO

Com base nos resultados obtidos, nos próximos dois subcapítulos, serão evidenciadas as conclusões mais pertinentes deste trabalho, as limitações encontradas ao longo de todos os meses do desenvolvimento do projeto e ainda sugestões para trabalhos futuros.

### 5.1. Conclusões finais

Como proposto, inicialmente foi realizada uma pesquisa bibliográfica majoritariamente sobre os diferentes métodos de monitorização da respiração. Associados a este tema, estão ligadas pequenas noções de anatomia e fisiologia do sistema respiratório e aplicação de filtros para processar o sinal respiratório.

A partir da pesquisa, foi possível reunir todas as informações necessárias para criar um sistema de monitorização de respiração para pacientes de radioterapia externa, em que é possível para os técnicos responsáveis controlar os diferentes momentos da respiração de cada doente, em cada sessão, e ainda passar uma informação diferente para o próprio paciente, informação que passaria por uma melhor interpretação, por parte do doente, da sua amplitude de respiração.

O marcador aplicado neste trabalho revelou-se mais útil que o que já era utilizado devido à sua versatilidade em relação ao vasto número de padrões existentes e o facto de que, após o seu reconhecimento por uma câmara, é possível criar um padrão respiratório fidedigno com base na movimentação em tempo real.

A deteção do marcador foi realizada com sucesso, demonstrando-se uma solução fácil e intuitiva de implementar. O processo de calibração também obteve bons resultados, revelando-se um processo importante para quantificar os movimentos realizados pelo marcador.

A linguagem de programação usada neste projeto, *Python*, acabou por ser uma excelente ferramenta de programação devido à sua simplicidade e fácil acesso. Além disso, é uma linguagem em consecutiva atualização e que facilmente permite a implementação de novas bibliotecas que possam surgir no futuro.

### 5.2. Limitações e investigação futura

Como qualquer projeto que envolva programação, a principal limitação deste trabalho foi o surgimento de erros durante a realização do código, ultrapassados posteriormente com a devida dedicação e ajuda externa (orientadores e páginas *web* próprias para resolução de erros de programação).

Como sugestão para trabalhos futuros, seria a continuação do trabalho realizado sobre o objeto 3D. Desenhar um modelo 3D do bloco colocado no abdómen do doente e, como parte integrante desse bloco, o desenho do padrão *ArUco*. Com esta implementação, já não seria necessário imprimir o padrão várias vezes, devido ao seu desgaste natural, pois já estaria implementado no bloco. Para além disso, acaba por ser uma solução muito mais barata para o instituto pois não teria de comprar os blocos com sensores, bastando apenas utilizar uma impressora 3D e já teria um novo objeto pronto a ser utilizado nos tratamentos.

A utilização de mais do que um marcador também seria uma ideia interessante para abordar futuramente, pois se forem utilizados em várias posições do corpo do doente, é possível obter mais informação sobre cada zona especificamente, obtendo uma monitorização mais completa.

Por fim, uma última sugestão para o futuro, seria a implementação deste sistema nas clínicas e, assim, expandindo esta ideia inovadora para mais locais, providenciando uma forma simples e eficaz de monitorizar e controlar os momentos de respiração de cada doente que seja submetido a sessões de radioterapia.

## REFERÊNCIAS BIBLIOGRÁFICAS

- [1] Mayo Clinic, “Radiation therapy.”
- [2] P. Giraud and A. Houle, “Respiratory Gating for Radiotherapy: Main Technical Aspects and Clinical Benefits,” *ISRN Pulmonol*, vol. 2013, pp. 1–13, Mar. 2013, doi: 10.1155/2013/519602.
- [3] Cleveland Clinic, “Respiratory System.”
- [4] Deborah Pedersen, Jessica Gotwals, and Tennille Dozier, “Anatomy of the Respiratory System.”
- [5] Elliot Carhart, “The Physiology of Respiration.”
- [6] E. Kaza *et al.*, “Lung volume reproducibility under ABC control and self-sustained breath-holding,” *J Appl Clin Med Phys*, vol. 18, no. 2, pp. 154–162, Mar. 2017, doi: 10.1002/acm2.12034.
- [7] Elekta, “ABC Full System.”
- [8] “Sistema de controle dos movimentos respiratórios Real-time Position Management™.”
- [9] P. Giraud and A. Houle, “Respiratory Gating for Radiotherapy: Main Technical Aspects and Clinical Benefits,” *ISRN Pulmonol*, vol. 2013, pp. 1–13, Mar. 2013, doi: 10.1155/2013/519602.
- [10] Elekta, “Active Breathing Coordinator™ and MotionView™ help to spare the heart and left lung in left breast cancer radiotherapy.”
- [11] Accuray, “Adapt and synchronize treatment delivery with Synchrony motion synchronization technology.”
- [12] Brainlab, “Precision Patient Monitoring for Radiotherapy.”
- [13] Philips, “Respiratory motion management for CT.”
- [14] Philips, “Brilliance CT Big Bore Oncology.”
- [15] “Surface Guided Radiation Therapy.”
- [16] H. Fayad, T. Pan, J. François Clement, and D. Visvikis, “Technical Note: Correlation of respiratory motion between external patient surface and internal anatomical landmarks,” *Med Phys*, vol. 38, no. 6Part1, pp. 3157–3164, May 2011, doi: 10.1118/1.3589131.
- [17] M. M. Suesada, H. de A. Carvalho, A. L. P. de Albuquerque, J. M. Salge, S. R. Stuart, and T. Y. Takagaki, “Impact of thoracic radiotherapy on respiratory function and exercise capacity in patients with breast cancer,” *Jornal Brasileiro de Pneumologia*, vol. 44, no. 6, pp. 469–476, Dec. 2018, doi: 10.1590/s1806-37562017000000120.
- [18] B. Elicker, C. A. de C. Pereira, R. Webb, and K. O. Leslie, “Padrões tomográficos das doenças intersticiais pulmonares difusas com correlação clínica e patológica,” *Jornal Brasileiro de Pneumologia*, vol. 34, no. 9, pp. 715–744, Sep. 2008, doi: 10.1590/S1806-37132008000900013.
- [19] T. Pulvirenti *et al.*, “Interfraction movement and clinical outcome of immobilization for thoracic irradiation: A randomized controlled trial,” *Precis Radiat Oncol*, vol. 2, no. 1, pp. 4–10, Mar. 2018, doi: 10.1002/pro6.35.
- [20] G. C. Bentel, L. B. Marks, and R. Krishnamurthy, “Impact of cradle immobilization on setup reproducibility during external beam radiation therapy for lung cancer,” Jun. 1997. doi: 10.1016/s0360-3016(97)00011-4.
- [21] P. Giraud *et al.*, “Conformal radiotherapy (CRT) planning for lung cancer: analysis of intrathoracic organ motion during extreme phases of breathing,” *Int J Radiat Oncol Biol Phys*, vol. 51, no. 4, pp. 1081–92, Nov. 2001, doi: 10.1016/s0360-3016(01)01766-7.
- [22] R. Kearvell and M. A. Ebert, “External respiratory motion for abdominal radiotherapy patients: implications for patient alignment,” *Med Dosim*, vol. 28, no. 4, pp. 217–22, 2003, doi: 10.1016/j.meddos.2003.08.002.
- [23] J. B. West, “The original presentation of Boyle’s law,” *J Appl Physiol (1985)*, vol. 87, no. 4, pp. 1543–5, Oct. 1999, doi: 10.1152/jappl.1999.87.4.1543.

- [24] J. A. Waxenbaum, V. Reddy, and B. Futterman, *Anatomy, Back, Thoracic Vertebrae*. 2023.
- [25] A. R. J, M. G, M. K, and R. P, “Closing the cleft over a throbbing heart: neonatal sternal cleft.,” *BMJ Case Rep*, vol. 2014, Jul. 2014, doi: 10.1136/bcr-2014-204529.
- [26] A.-H. Monsoro-Burq, “Sclerotome development and morphogenesis: when experimental embryology meets genetics.,” *Int J Dev Biol*, vol. 49, no. 2–3, pp. 301–8, 2005, doi: 10.1387/ijdb.041953am.
- [27] C. Ravi and C. L. McKnight, *Chest Tube*. 2023.
- [28] K. Davignon, J. Kwo, and L. M. Bigatello, “Pathophysiology and management of the flail chest.,” *Minerva Anesthesiol*, vol. 70, no. 4, pp. 193–9, Apr. 2004.
- [29] S. Fleming *et al.*, “Normal ranges of heart rate and respiratory rate in children from birth to 18 years of age: a systematic review of observational studies.,” *Lancet*, vol. 377, no. 9770, pp. 1011–8, Mar. 2011, doi: 10.1016/S0140-6736(10)62226-X.
- [30] A. Rodríguez-Molinero, L. Narvaiza, J. Ruiz, and C. Gálvez-Barrón, “Normal respiratory rate and peripheral blood oxygen saturation in the elderly population.,” *J Am Geriatr Soc*, vol. 61, no. 12, pp. 2238–2240, Dec. 2013, doi: 10.1111/jgs.12580.
- [31] Alene Burke, “Respiratory System: TEAS.”
- [32] M. Brink, C. H. Müller, and C. Schierz, “Contact-free measurement of heart rate, respiration rate, and body movements during sleep,” *Behav Res Methods*, vol. 38, no. 3, pp. 511–521, Aug. 2006, doi: 10.3758/BF03192806.
- [33] Z. T. Beattie, C. C. Hagen, M. Pavel, and T. L. Hayes, “Classification of breathing events using load cells under the bed,” in *2009 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, IEEE, Sep. 2009, pp. 3921–3924. doi: 10.1109/IEMBS.2009.5333548.
- [34] Joonas Paalasmaa, “Monitoring Sleep with Force Sensor Measurement,” Feb. 2014.
- [35] Antti Vehkaoja, “Unobtrusive Monitoring of Heart Rate and Respiration Rate during Sleep,” Feb. 2015.
- [36] W. Lee, H. Yoon, C. Han, K. Joo, and K. Park, “Physiological Signal Monitoring Bed for Infants Based on Load-Cell Sensors,” *Sensors*, vol. 16, no. 3, p. 409, Mar. 2016, doi: 10.3390/s16030409.
- [37] “High-pass filtering.”
- [38] “Tudo sobre Python: o que é? Para que serve? Como funciona? Saiba mais.”
- [39] Daniel Flohr and Jan Fischer, “A Lightweight ID-Based Extension for Marker Tracking Systems,” Jan. 2007.
- [40] William C. Evans, “Exponentially-Weighted Moving Average.”

## APÊNDICE A

### Código em *Python*

```
project.py > ...
1  import cv2
2  import numpy as np
3  import matplotlib.pyplot as plt
4  from matplotlib.backends.backend_tkagg import FigureCanvasTkAgg
5  from matplotlib.figure import Figure
6  import tkinter as tk
7  from PIL import Image, ImageTk
8  import os
9  from scipy.signal import butter, filtfilt, lfilter
10 from sklearn.preprocessing import MinMaxScaler
11 from mplcursors import Selection
12 from tkinter import Scale
13 import sqlite3
14 import time
15 import subprocess
16 from scipy.signal import hilbert
17 from scipy.signal import find_peaks
18 import scipy.signal as signal
19 from scipy.signal import argrelextrema
20 from scipy.signal import savgol_filter
21 from tkinter import messagebox
22 import sys
23
24 # Inicialização da janela Tkinter
25 root = tk.Tk()
26 root.title("Monitorização da Respiração")
27 root.geometry("800x600")
28 root.configure(bg="white")
29
30 root1 = tk.Tk()
31 root1.title("Lista de Pacientes")
32 root1.geometry("400x400")
33
34 # Variáveis globais
35 breath_data = []
36 breath_smoothed = []
37 breath_curves = []
```

Figura 22: Código Principal Parte 1

```

project.py > ...
35 breath_data = []
36 breath_smoothed = []
37 breath_curves = []
38 previous_marker_position = None
39 is_upward = False # Flag para indicar se o padrão está a subir
40 pattern_top_position = None # Posição do padrão quando ele está no topo
41 initial_amplitude = None
42 previous_max = None
43 breath_phase = 0.0
44 time_elapsed = 0
45 base_position = None
46 previous_movement = None
47 pattern_detected = False
48 is_pattern_visible = False
49 last_value = 0
50 last_pattern_value = 0
51 # Variável global para indicar se a detecção de padrão está ativa
52 pattern_detection_active = False
53 tratamento_detection_active = False
54 current_baseline = 0
55 middle_mouse_button_pressed = False
56 selected_patient_id = None
57 selected_patient_name = None
58 selected_patient_number = None
59
60
61 calib_data = np.load("D:/ISEP/Engenharia_Biomedica/Mestrado/2_ano/treino/calib_data/MultiMatrix.npz")
62 cam_matrix = calib_data["camMatrix"]
63 dist_coef = calib_data["distCoef"]
64
65 # Função para detecção de marcadores ArUco
66 def detect_aruco_markers():
67     global previous_marker_position
68     global breath_smoothed
69     global is_upward
70     global pattern_top_position
71     global breath_data
72     global time_elapsed

```

Figura 23: Código Principal Parte 2

```

project.py > ...
70     global pattern_top_position
71     global breath_data
72     global time_elapsed
73     global pattern_detection_active
74
75     # Captura da imagem da webcam
76     ret, frame = cap.read()
77     if not ret:
78         return
79
80     frame_undistorted = cv2.undistort(frame, cam_matrix, dist_coef)
81
82     # Conversão para escala de cinza
83     gray = cv2.cvtColor(frame_undistorted, cv2.COLOR_BGR2GRAY)
84
85     # Ajuste e Detecção dos marcadores ArUco
86     parameters = cv2.aruco.DetectorParameters()
87     parameters.minMarkerPerimeterRate = 0.03
88     parameters.cornerRefinementMethod = cv2.aruco.CORNER_REFINE_SUBPIX
89     parameters.adaptiveThreshWinSizeMin = 3
90     parameters.adaptiveThreshWinSizeMax = 23
91     corners, ids, _ = cv2.aruco.detectMarkers(gray, aruco_dict, parameters=parameters)
92
93     # Rastreamento dos marcadores
94     if ids is not None:
95         cv2.aruco.drawDetectedMarkers(frame, corners)
96
97         # Cálculo da amplitude do movimento da respiração
98         if len(corners) > 0:
99             current_marker_position = corners[0][0][1] # Coordenada do canto inferior esquerdo do marcador
100
101             if pattern_top_position is None:
102                 pattern_top_position = current_marker_position[1] # Define a posição inicial do topo do padrão
103
104             # Cálculo da diferença vertical entre a posição atual e a posição do topo do padrão como amplitude
105             breath_amplitude = pattern_top_position - current_marker_position[1]
106

```

Figura 24: Código Principal Parte 3

```
project.py > ...
105     breath_amplitude = pattern_top_position - current_marker_position[1]
106
107     # Atualização do gráfico apenas se a amplitude for maior que um determinado valor
108     if abs(breath_amplitude) > 5: # Valor de amplitude arbitrário para considerar o movimento ascendente
109         is_upward = True if breath_amplitude > 0 else False
110
111         # Cálculo do fator de escala para ajustar a curva sinusoidal
112         scale_factor = 1.0 + (breath_amplitude / 100.0)
113
114         # Atualização do gráfico
115         breath_data.append(scale_factor) # Atualização dos dados do gráfico
116
117         breath_smoothed = apply_ewma_filter(breath_data, alpha=0.4)
118         update_breath_graph(breath_smoothed)
119
120     else:
121         # Incremento do tempo decorrido mesmo quando o padrão não é mostrado
122         time_elapsed += 1
123
124         # Adicionar valor zero para indicar a ausência de padrão
125         breath_data.append(0)
126         breath_smoothed = apply_ewma_filter(breath_data, alpha=0.4)
127
128         # Substituir o último valor pela versão suavizada
129         breath_data[-1] = breath_smoothed[-1]
130
131         update_breath_graph(breath_smoothed)
132
133     # Exibição do frame com os marcadores
134     frame = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
135     img = Image.fromarray(frame)
136     img = img.resize((320, 240)) # Redimensiona o tamanho da imagem
137     imgtk = ImageTk.PhotoImage(image=img)
138     video_label.configure(image=imgtk)
139     video_label.image = imgtk
140
141     if pattern_detection_active:
```

Figura 25: Código Principal Parte 4



```
project.py > ...
140
141     if pattern_detection_active:
142         video_label.after(1, detect_aruco_markers)
143
144     # Função para iniciar ou parar a detecção de padrão
145     def toggle_pattern_detection():
146         global pattern_detection_active
147         pattern_detection_active = not pattern_detection_active
148         if pattern_detection_active:
149             detect_aruco_markers()
150
151     # Função para aplicar filtro EWMA e suavizar o sinal
152     def apply_ewma_filter(data, alpha):
153         filtered_data = [data[0]] # O primeiro valor é igual ao valor original
154         for i in range(1, len(data)):
155             filtered_value = alpha * data[i] + (1 - alpha) * filtered_data[i-1]
156             filtered_data.append(filtered_value)
157         return np.array(filtered_data)
158
159     # Variáveis globais para armazenar os valores dos limites
160     upper_limit = 0
161     lower_limit = 0
162     baseline = 0
163     upper_limit_line = None
164     lower_limit_line = None
165     baseline_line = None
166     baseline_dragging = False
167     current_action = None
168     adjusted_limits_saved = False
169     adjusted_upper_limit = 0
170     adjusted_lower_limit = 0
171     upper_trat = 0
172     lower_trat = 0
173     baseline_to_upper_limit = 0
174     baseline_to_lower_limit = 0
175
176
```

Figura 26: Código Principal Parte 5

```
project.py > ...
177 # Função para atualizar o gráfico da respiração
178 def update_breath_graph(breath_smoothed):
179
180     global upper_limit
181     global lower_limit
182     global upper_limit_line
183     global lower_limit_line
184     global baseline_line
185     global baseline_dragging
186     global is_pattern_visible
187     global last_value
188     global baseline
189     global last_pattern_value
190     global adjusted_lower_limit
191     global adjusted_upper_limit
192     global selected_patient_number
193     global selected_patient_id
194     global baseline_to_upper_limit
195     global baseline_to_lower_limit
196
197
198     fig.clear()
199     plot = fig.add_subplot(1, 1, 1)
200
201     # Obter os valores mínimos e máximos da curva suavizada
202     min_value = np.min(breath_smoothed)
203     max_value = np.max(breath_smoothed)
204
205     """ # Se o valor mínimo for menor que zero, ajuste a curva para começar em zero
206     if min_value < 0:
207         breath_smoothed_adjusted = breath_smoothed - min_value
208         max_value -= min_value
209     else:
210         breath_smoothed_adjusted = breath_smoothed """
211
212     # Calcular o eixo de tempo em segundos
213     sample_rate = 10 # Taxa de amostragem (1 amostra por segundo)
```

Figura 27: Código Principal Parte 6

```
project.py > ...
212     # Calcular o eixo de tempo em segundos
213     sample_rate = 10 # Taxa de amostragem (1 amostra por segundo)
214     num_samples = len(breath_smoothed)
215     time_axis = np.arange(num_samples) / sample_rate
216     time_range_seconds = 40
217
218     # Manter apenas os últimos 40 segundos de dados
219     max_samples = time_range_seconds * sample_rate
220     if num_samples > max_samples:
221         breath_smoothed_adjusted = breath_smoothed[-max_samples:]
222         time_axis = time_axis[-max_samples:]
223     else:
224         breath_smoothed_adjusted = breath_smoothed
225
226
227     plot.plot(time_axis, breath_smoothed_adjusted, 'b-')
228
229     plt.style.use('ggplot')
230
231     # Adicionar a área preenchida entre os limites inferior e superior
232     plot.fill_between(time_axis, lower_limit, upper_limit, color='green', alpha=0.3)
233
234     # Definir limites máximos e mínimos para a escala do eixo y
235     y_lower_limit = -0.5
236     y_upper_limit = max_value * 1.1
237
238     # Ajustar os limites da escala do eixo y
239     plot.set_ylim([y_lower_limit, y_upper_limit])
240
241     # Definir limites máximos e mínimos para a escala do eixo x
242     x_lower_limit = time_axis[0] # Começar no tempo da primeira amostra
243     x_upper_limit = time_axis[-1] # Terminar no tempo da última amostra
244
245     # Ajustar os limites da escala do eixo x
246     plot.set_xlim([x_lower_limit, x_upper_limit])
247
248     plot.set_xlabel("Tempo (s)", fontsize=10)
```

Figura 28: Código Principal Parte 7

```
project.py > ...
248 plot.set_xlabel("Tempo (s)", fontsize=10)
249 plot.set_ylabel("Amplitude (cm)", fontsize=10)
250 plot.set_title("Padrão de Respiração", fontsize=16)
251 plot.grid(True, linestyle='--', alpha=0.5, color='lightgrey')
252 plot.legend(fontsize=10)
253
254 # Criar a linha da baseline no gráfico
255 baseline_line = plot.axhline(baseline, color='m', linestyle='--')
256
257
258 upper_limit_line = plot.axhline(upper_limit, color='g', linestyle='--')
259 lower_limit_line = plot.axhline(lower_limit, color='r', linestyle='--')
260
261 def on_move(event):
262     if event.inaxes == plot:
263         x, y = event.xdata, event.ydata
264         global upper_limit
265         global lower_limit
266         global baseline
267         global baseline_line
268         global middle_mouse_button_pressed
269         global baseline_to_upper_limit
270         global baseline_to_lower_limit
271
272         if event.button == 1: # Botão esquerdo do rato para mover o limite superior
273             upper_limit = y
274             upper_limit_line.set_ydata([upper_limit, upper_limit])
275             # Atualizar a diferença entre a baseline e o limite superior
276             baseline_to_upper_limit = upper_limit - baseline
277         elif event.button == 3: # Botão direito do rato para mover o limite inferior
278             lower_limit = y
279             lower_limit_line.set_ydata([lower_limit, lower_limit])
280             # Atualizar a diferença entre a baseline e o limite inferior
281             baseline_to_lower_limit = baseline - lower_limit
282         elif event.button == 2: # Botão do meio do rato para mover a baseline
283             baseline = y
284             baseline_line.set_ydata([baseline, baseline])
```

Figura 29: Código Principal Parte 8

```

project.py > ...
284     baseline_line.set_ydata([baseline, baseline])
285     # Atualizar os limites com base na nova posição da baseline
286     upper_limit = baseline + baseline_to_upper_limit
287     lower_limit = baseline - baseline_to_lower_limit
288     upper_limit_line.set_ydata([upper_limit, upper_limit])
289     lower_limit_line.set_ydata([lower_limit, lower_limit])
290
291     fig.canvas.draw()
292
293     fig.canvas.mpl_connect('button_press_event', on_move)
294     fig.canvas.mpl_connect('motion_notify_event', on_move)
295
296     # Adicionar rótulos de texto para os limites e a baseline
297     text_upper_limit = f'LS: {upper_limit:.2f}'
298     text_lower_limit = f'LI: {lower_limit:.2f}'
299     text_baseline = f'Baseline: {baseline:.2f}'
300
301     # Adicionar rótulos de texto separados para os limites e a baseline
302     upper_limit_label = tk.Label(root, text=text_upper_limit)
303     lower_limit_label = tk.Label(root, text=text_lower_limit)
304     baseline_label = tk.Label(root, text=text_baseline)
305
306     # Posicionar os rótulos de texto
307     upper_limit_label.place(x=1080, y=300)
308     lower_limit_label.place(x=1080, y=320)
309     baseline_label.place(x=1080, y=340)
310
311     # Chama a função para atualizar a imagem com base nos limites
312     update_image(lower_limit, upper_limit, breath_smoothed_adjusted)
313
314     # Atualização do gráfico na interface Tkinter
315     canvas.draw()
316
317     def update_image(lower_limit, upper_limit, breath_smoothed):
318         if lower_limit <= breath_smoothed[-1] <= upper_limit:
319             image_label.configure(image=image_within_limits)
320         else:

```

Figura 30: Código Principal Parte 9

```
project.py > ...
320     else:
321         image_label.configure(image=image_outside_limits)
322
323     image_label.image = image_label.cget("image") # Atualizar a imagem mostrada
324
325
326 def calibration():
327     # Obter o diretório atual do projeto principal
328     project_dir = os.path.dirname(os.path.abspath(__file__))
329
330     # Construir os caminhos absolutos para cada ficheiro
331     capture_images_py_path = os.path.join(project_dir, "camera_calibration", "capture_images.py")
332     calibration_script_py_path = os.path.join(project_dir, "camera_calibration", "calibration_script.py")
333     check_py_path = os.path.join(project_dir, "distance_check", "check.py")
334
335     # Executar os ficheiros usando os caminhos absolutos
336     subprocess.run(["python", capture_images_py_path])
337     subprocess.run(["python", calibration_script_py_path])
338     subprocess.run(["python", check_py_path])
339
340 # Function to open the new patient window
341 def open_new_patient_window():
342     new_patient_window = tk.Toplevel(root1)
343     new_patient_window.title("Adicionar Novo Paciente")
344
345     # Add input fields for patient information
346     tk.Label(new_patient_window, text="Nome do Paciente:").pack()
347     patient_name_entry = tk.Entry(new_patient_window)
348     patient_name_entry.pack()
349
350     tk.Label(new_patient_window, text="OBS do Paciente:").pack()
351     patient_number_entry = tk.Entry(new_patient_window)
352     patient_number_entry.pack()
353
354 # Function to add the new patient to the database
355 def add_patient():
356     name = patient_name_entry.get()
```

Figura 31: Código Principal Parte 10

```
project.py > ...
355 def add_patient():
356     name = patient_name_entry.get()
357     number = patient_number_entry.get()
358     if name and number:
359         cursor.execute("INSERT INTO patients (name, number) VALUES (?, ?)", (name, number))
360         conn.commit()
361         update_patient_list()
362         new_patient_window.destroy()
363     else:
364         messagebox.showerror("Erro", "Por favor, preencha todos os campos.")
365
366     tk.Button(new_patient_window, text="Adicionar Paciente", command=add_patient).pack()
367
368 # Function to update the patient list in the Listbox
369 def update_patient_list():
370     patients_listbox.delete(0, tk.END)
371     cursor.execute("SELECT * FROM patients")
372     rows = cursor.fetchall()
373     for row in rows:
374         patients_listbox.insert(tk.END, f"N: {row[1]} | OBS: {row[2]}")
375
376 def select_patient(event):
377     global selected_patient_id, selected_patient_name, selected_patient_number
378     selected_index = patients_listbox.curselection()
379     if selected_index:
380         selected_index = selected_index[0]
381         selected_patient_id = rows[selected_index][0]
382         selected_patient_name = rows[selected_index][1]
383         selected_patient_number = rows[selected_index][2]
384
385         # Abrir a janela "root" associada ao paciente selecionado
386         open_root_window()
387
388     patients_listbox.bind("<<ListboxSelect>>", select_patient)
389
390 def open_root_window():
391     global selected_patient_id
```

Figura 32: Código Principal Parte 11

```

project.py > ...
390 def open_root_window():
391     global selected_patient_id
392     global selected_patient_name
393     global selected_patient_number
394     if selected_patient_id is not None:
395         # Abre a janela "root" e configura os elementos da interface com as informações do paciente
396         root.title(f"Monitorização da Respiração - Paciente: {selected_patient_name} ({selected_patient_number})")
397
398         root.deiconify() # Torna a janela "root" visível
399
400 def write_data_to_file(file_name, time_points, breath_data):
401     with open(file_name, "w") as file:
402         for time_point, value in zip(time_points, breath_data):
403             file.write(f"Tempo(s): {time_point/10}, Amplitude(cm): {value}\n")
404
405 def perform_tac():
406     global breath_smoothed, selected_patient_id, selected_patient_number
407
408     if selected_patient_id is not None:
409         # Nome do arquivo de dados
410         current_date = time.strftime("%Y-%m-%d")
411         file_name = f"TAC_{current_date}_{selected_patient_number}.txt"
412
413         # Usa os valores reais de tempo e amplitude suavizada
414         time_points = np.arange(len(breath_smoothed))
415         write_data_to_file(file_name, time_points, breath_smoothed)
416
417         messagebox.showinfo("Sucesso", f"Dados TAC guardados em '{file_name}'.")
418     else:
419         messagebox.showerror("Erro", "Selecione um paciente antes de guardar os dados TAC.")
420
421 def perform_treatment():
422     global breath_smoothed, selected_patient_id, selected_patient_number
423
424     if selected_patient_id is not None:
425         # Nome do arquivo de dados
426         current_date = time.strftime("%Y-%m-%d")

```

Figura 33: Código Principal Parte 12

```

project.py > ...
426     current_date = time.strftime("%Y-%m-%d")
427     file_name = f"Tratamento_{current_date}_{selected_patient_number}.txt"
428
429     # Usa os valores reais de tempo e amplitude suavizada
430     time_points = np.arange(len(breath_smoothed))
431     write_data_to_file(file_name, time_points, breath_smoothed)
432
433     messagebox.showinfo("Sucesso", f"Dados Tratamento guardados em '{file_name}'.")
434 else:
435     messagebox.showerror("Erro", "Selecione um paciente antes de guardar os dados Tratamento.")
436
437 def save_limits():
438     global selected_patient_id
439     global upper_limit
440     global lower_limit
441
442     if selected_patient_id is not None:
443         cursor.execute("UPDATE patients SET upper_limit = ?, lower_limit = ? WHERE id = ?",
444                       (upper_limit, lower_limit, selected_patient_id))
445         conn.commit()
446         messagebox.showinfo("Sucesso", "Limites TAC guardados com sucesso.")
447     else:
448         messagebox.showerror("Erro", "Selecione um paciente antes de guardar os limites TAC.")
449
450 def show_patient_limits():
451     global selected_patient_id, selected_patient_name, selected_patient_number
452     global baseline
453
454     # Verifique se algum paciente foi selecionado
455     if selected_patient_id is not None:
456         # Consulta SQL para obter os limites do paciente selecionado
457         cursor.execute("SELECT upper_limit, lower_limit FROM patients WHERE id = ? OR name = ? OR number = ?",
458                       (selected_patient_id, selected_patient_name, selected_patient_number))
459         patient_limits = cursor.fetchone()
460
461         if patient_limits:
462             upper_limit, lower_limit = patient_limits

```

Figura 34: Código Principal Parte 13



```

project.py > ...
462     upper_limit, lower_limit = patient_limits
463     info_text = f"Limite Superior: {upper_limit: .2f}\nLimite Inferior: {lower_limit: .2f}"
464     info_label.config(text=info_text)
465     else:
466         info_label.config(text="Limite superior e inferior não encontrados para o paciente selecionado.")
467     else:
468         info_label.config(text="Nenhum paciente selecionado.")
469
470 # Create or connect to the SQLite database
471 conn = sqlite3.connect("patient_database.db")
472 cursor = conn.cursor()
473
474 # Create the patients table if it doesn't exist
475 cursor.execute("""
476     CREATE TABLE IF NOT EXISTS patients (
477         id INTEGER PRIMARY KEY AUTOINCREMENT,
478         name TEXT NOT NULL,
479         number TEXT NOT NULL,
480         upper_limit REAL,
481         lower_limit REAL
482     )
483 """)
484 conn.commit()
485
486
487 # Create a Listbox to display the patient list
488 patients_listbox = tk.Listbox(root1)
489 patients_listbox.pack()
490
491 # Create a button to add new patients
492 add_patient_button = tk.Button(root1, text="Adicionar Novo Paciente", command=open_new_patient_window)
493 add_patient_button.pack()
494
495 # Inicialização da captura de vídeo
496 cap = cv2.VideoCapture(0)
497
498 # Carregamento do dicionário ArUco

```

Figura 35: Código Principal Parte 14

```

project.py > ...
498 # Carregamento do dicionário ArUco
499 aruco_dict = cv2.aruco.getPredefinedDictionary(cv2.aruco.DICT_6X6_250)
500
501 # Carregar as imagens usando a biblioteca PIL
502 image_within_limits = Image.open("beam_on.png")
503 image_outside_limits = Image.open("beam_off.png")
504
505 # Redimensionar as imagens para o tamanho desejado
506 image_within_limits = image_within_limits.resize((200, 200))
507 image_outside_limits = image_outside_limits.resize((200, 200))
508
509 # Converter as imagens para o formato Tkinter
510 image_within_limits = ImageTk.PhotoImage(image_within_limits)
511 image_outside_limits = ImageTk.PhotoImage(image_outside_limits)
512
513 # Criação do quadro para a câmara
514 video_frame = tk.Frame(root, width=160, height=120)
515 video_frame.grid(row=0, column=0, padx=10, pady=10)
516 video_label = tk.Label(video_frame)
517 video_label.pack()
518
519 # Criação do quadro para o gráfico
520 graph_frame = tk.Frame(root)
521 graph_frame.grid(row=1, column=2, padx=10, pady=10)
522
523 # Criação dos rótulos para exibir as imagens
524 image_label = tk.Label(root, image=image_within_limits)
525 image_label.grid(row=1, column=0, padx=10, pady=10)
526
527 # Criação dos botões
528 button_frame = tk.Frame(root)
529 button_frame.grid(row=0, column=3, padx=10, pady=10)
530
531 detection_frame = tk.Frame(root)
532 detection_frame.grid(row=1, column=1, padx=10, pady=10)
533
534 button1_frame = tk.Frame(root)

```

Figura 36: Código Principal Parte 15

```
project.py > ...
533
534 button1_frame = tk.Frame(root)
535 button1_frame.grid(row=1, column=3, padx=10, pady=10)
536
537 save_frame = tk.Frame(root)
538 save_frame.grid(row=0, column=2, padx=10, pady=10)
539
540 # Criação do botão para iniciar/parar a detecção de padrão
541 pattern_detection_button = tk.Button(detection_frame, text="Iniciar Detecção", command=toggle_pattern_detection)
542 pattern_detection_button.pack()
543
544 tac_button = tk.Button(button_frame, text="Guardar Referência", command=perform_tac)
545 tac_button.pack()
546
547 tratamento_button = tk.Button(button1_frame, text="Guardar Tratamento", command=perform_treatment)
548 tratamento_button.pack()
549
550 save_button = tk.Button(save_frame, text="Guardar Limites de Referência", command=save_limits)
551 save_button.pack()
552
553 # Criar um botão para mostrar informações do paciente selecionado
554 info_button = tk.Button(root, text="Mostrar Limites do Paciente", command=show_patient_limits)
555 info_button.grid(row=0, column=1, padx=10, pady=10)
556
557 # Criar uma etiqueta para mostrar as informações do paciente
558 info_label = tk.Label(root, text="")
559 info_label.grid(row=0, column=1, padx=10, pady=10)
560
561 # Criação da figura para o gráfico da respiração
562 fig = Figure(figsize=(6, 5), dpi=100)
563 canvas = FigureCanvasTkAgg(fig, master=graph_frame)
564 canvas.get_tk_widget().pack()
565
566 calibration()
567
568 update_patient_list()
```

Figura 37: Código Principal Parte 16

```
565
566 calibration()
567
568 update_patient_list()
569
570 root1.mainloop()
571 conn.close()
572
573 cap.release()
574 cv2.destroyAllWindows()
575
```

Figura 38: Código Principal Parte 17

```
camera_calibration > capture_images.py > ...
1  import cv2 as cv
2  import os
3
4  Chess_Board_Dimensions = (9, 6)
5
6  n = 0 # image counter
7
8  # checks images dir is exist or not
9  image_path = "images"
10
11 Dir_Check = os.path.isdir(image_path)
12
13 if not Dir_Check: # if directory does not exist, a new one is created
14     os.makedirs(image_path)
15     print(f"{image_path}" Directory is created')
16 else:
17     print(f"{image_path}" Directory already exists.')
18
19 criteria = (cv.TERM_CRITERIA_EPS + cv.TERM_CRITERIA_MAX_ITER, 30, 0.001)
20
21
22 def detect_checker_board(image, grayImage, criteria, boardDimension):
23     ret, corners = cv.findChessboardCorners(grayImage, boardDimension)
24     if ret == True:
25         corners1 = cv.cornerSubPix(grayImage, corners, (3, 3), (-1, -1), criteria)
26         image = cv.drawChessboardCorners(image, boardDimension, corners1, ret)
27
28     return image, ret
29
30
31 cap = cv.VideoCapture(1)
32
33 while True:
34     _, frame = cap.read()
35     copyFrame = frame.copy()
36     gray = cv.cvtColor(frame, cv.COLOR_BGR2GRAY)
37
```

Figura 39: Código da Calibração Parte 1

```
camera_calibration > capture_images.py > ...
35     copyFrame = frame.copy()
36     gray = cv.cvtColor(frame, cv.COLOR_BGR2GRAY)
37
38     image, board_detected = detect_checker_board(
39         frame, gray, criteria, Chess_Board_Dimensions
40     )
41     # print(ret)
42     cv.putText(
43         frame,
44         f"saved_img : {n}",
45         (30, 40),
46         cv.FONT_HERSHEY_PLAIN,
47         1.4,
48         (0, 255, 0),
49         2,
50         cv.LINE_AA,
51     )
52
53     cv.imshow("frame", frame)
54     # copyframe; without augmentation
55     cv.imshow("copyFrame", copyFrame)
56
57     key = cv.waitKey(1)
58
59     if key == ord("q"):
60         break
61     if key == ord("s") and board_detected == True:
62         # the checker board image gets stored
63         cv.imwrite(f"{image_path}/image{n}.png", copyFrame)
64
65         print(f"saved image number {n}")
66         n += 1 # the image counter: incrementing
67 cap.release()
68 cv.destroyAllWindows()
69
70 print("Total saved Images:", n)
```

Figura 40: Código da Calibração Parte 2

```
camera_calibration > calibration_script.py > ...
1  import cv2 as cv
2  import os
3  import numpy as np
4
5  # Chess/checker board size, dimensions
6  CHESS_BOARD_DIM = (9, 6)
7
8  # The size of squares in the checker board design.
9  SQUARE_SIZE = 22 # millimeters (change it according to printed size)
10
11 # termination criteria
12 criteria = (cv.TERM_CRITERIA_EPS + cv.TERM_CRITERIA_MAX_ITER, 30, 0.001)
13
14
15 calib_data_path = "../calib_data"
16 CHECK_DIR = os.path.isdir(calib_data_path)
17
18
19 # saving the image / camera calibration data
20
21 if not CHECK_DIR:
22     os.makedirs(calib_data_path)
23     print(f"{calib_data_path}" Directory is created")
24
25 else:
26     print(f"{calib_data_path}" Directory already Exists.")
27
28 # prepare object points, i.e. (0,0,0), (1,0,0), (2,0,0) ..., (6,5,0)
29 obj_3D = np.zeros((CHESS_BOARD_DIM[0] * CHESS_BOARD_DIM[1], 3), np.float32)
30
31 obj_3D[:, :2] = np.mgrid[0 : CHESS_BOARD_DIM[0], 0 : CHESS_BOARD_DIM[1]].T.reshape(
32     -1, 2
33 )
34 obj_3D *= SQUARE_SIZE
35 print(obj_3D)
36
```

Figura 41: Código da Calibração Parte 3

```
camera_calibration > calibration_script.py > ...
37 # Arrays to store object points and image points from all the given images.
38 obj_points_3D = [] # 3d point in real world space
39 img_points_2D = [] # 2d points in image plane
40
41 # The images directory path
42 image_dir_path = "images"
43
44 files = os.listdir(image_dir_path) # list of names of all the files present
45 for file in files:
46     print(file)
47     imagePath = os.path.join(image_dir_path, file)
48     # print(imagePath)
49
50     image = cv.imread(imagePath)
51     grayScale = cv.cvtColor(image, cv.COLOR_BGR2GRAY)
52     ret, corners = cv.findChessboardCorners(image, CHESS_BOARD_DIM, None)
53     if ret == True:
54         obj_points_3D.append(obj_3D)
55         corners2 = cv.cornerSubPix(grayScale, corners, (3, 3), (-1, -1), criteria)
56         img_points_2D.append(corners2)
57
58         img = cv.drawChessboardCorners(image, CHESS_BOARD_DIM, corners2, ret)
59
60 cv.destroyAllWindows()
61 # h, w = image.shape[:2]
62 ret, mtx, dist, rvecs, tvecs = cv.calibrateCamera(
63     obj_points_3D, img_points_2D, grayScale.shape[::-1], None, None
64 )
65 print("calibrated")
66
67 print("dumping the data into one files using numpy ")
68 np.savez(
69     f"{calib_data_path}/MultiMatrix",
70     camMatrix=mtx,
71     distCoef=dist,
72     rVector=rvecs,
73     tVector=tvecs,
```

Figura 42: Código da Calibração Parte 4

```

68     np.savez(
69         f"{calib_data_path}/MultiMatrix",
70         camMatrix=mtx,
71         distCoef=dist,
72         rVector=rvecs,
73         tVector=tvecs,
74     )
75
76     print("-----")
77
78     print("loading data stored using numpy savez function\n \n \n")
79
80     data = np.load(f"{calib_data_path}/MultiMatrix.npz")
81
82     camMatrix = data["camMatrix"]
83     distCof = data["distCoef"]
84     rVector = data["rVector"]
85     tVector = data["tVector"]
86
87     print("loaded calibration data successfully")

```

Figura 43: Código da Calibração Parte 5

```

distance_check > check.py > ...
1  import cv2 as cv
2  from cv2 import aruco
3  import numpy as np
4
5  # load in the calibration data
6  calib_data_path = "../calib_data/MultiMatrix.npz"
7
8  calib_data = np.load(calib_data_path)
9  print(calib_data.files)
10
11  cam_mat = calib_data["camMatrix"]
12  dist_coef = calib_data["distCoef"]
13  r_vectors = calib_data["rVector"]
14  t_vectors = calib_data["tVector"]
15
16  MARKER_SIZE = 3 # centimeters (measure your printed marker size)
17
18  marker_dict = aruco.getPredefinedDictionary(aruco.DICT_6X6_250)
19
20  param_markers = aruco.DetectorParameters()
21
22  cap = cv.VideoCapture(1)
23
24  while True:
25      ret, frame = cap.read()
26      if not ret:
27          break
28      gray_frame = cv.cvtColor(frame, cv.COLOR_BGR2GRAY)
29      marker_corners, marker_ids, reject = aruco.detectMarkers(
30          gray_frame, marker_dict, parameters=param_markers
31      )
32      if marker_corners:
33          rVec, tVec, _ = aruco.estimatePoseSingleMarkers(
34              marker_corners, MARKER_SIZE, cam_mat, dist_coef
35          )
36          total_markers = range(0, marker_ids.size)
37          for ids, corners, i in zip(marker_ids, marker_corners, total_markers):

```

Figura 44: Código da Calibração Parte 6

```

distance_check > check.py > ...
36 total_markers = range(0, marker_ids.size)
37 for ids, corners, i in zip(marker_ids, marker_corners, total_markers):
38     cv.polylines(
39         frame, [corners.astype(np.int32)], True, (0, 255, 255), 4, cv.LINE_AA
40     )
41     corners = corners.reshape(4, 2)
42     corners = corners.astype(int)
43     top_right = corners[0].ravel()
44     top_left = corners[1].ravel()
45     bottom_right = corners[2].ravel()
46     bottom_left = corners[3].ravel()
47
48     # Calculating the distance
49     distance = np.sqrt(
50         tVec[i][0][2] ** 2 + tVec[i][0][0] ** 2 + tVec[i][0][1] ** 2
51     )
52     # Draw the pose of the marker
53     point = cv.drawFrameAxes(frame, cam_mat, dist_coef, rVec[i], tVec[i], 4, 4)
54     cv.putText(
55         frame,
56         f"id: {ids[0]} Dist: {round(distance, 2)}",
57         top_right,
58         cv.FONT_HERSHEY_PLAIN,
59         1.3,
60         (0, 0, 255),
61         2,
62         cv.LINE_AA,
63     )
64     cv.putText(
65         frame,
66         f"x:{round(tVec[i][0][0],1)} y: {round(-tVec[i][0][1],1)} ",
67         bottom_right,
68         cv.FONT_HERSHEY_PLAIN,
69         1.0,
70         (0, 0, 255),
71         2,
72         cv.LINE_AA,
73     )

```

Figura 45: Código da Calibração Parte 7

```

68         cv.FONT_HERSHEY_PLAIN,
69         1.0,
70         (0, 0, 255),
71         2,
72         cv.LINE_AA,
73     )
74     # print(ids, " ", corners)
75     cv.imshow("frame", frame)
76     key = cv.waitKey(1)
77     if key == ord("q"):
78         break
79 cap.release()
80 cv.destroyAllWindows()

```

Figura 46: Código da Calibração Parte 8