



Universidade Estadual de Feira de Santana
Programa de Pós-Graduação em Computação Aplicada

Maximização da Cobertura Redundante em Redes de Sensores Visuais Sem Fio para Monitoramento de Alvos

Tassalon Ferreira da Silva

Feira de Santana

2016



Universidade Estadual de Feira de Santana
Programa de Pós-Graduação em Computação Aplicada

Tassalon Ferreira da Silva

**Maximização da Cobertura Redundante em
Redes de Sensores Visuais Sem Fio para
Monitoramento de Alvos**

Dissertação apresentada à Universidade Estadual de Feira de Santana como parte dos requisitos para a obtenção do título de Mestre em Computação Aplicada.

Orientador: Dr. Daniel Gouveia Costa

Feira de Santana

2016

Ficha Catalográfica – Biblioteca Central Julieta Carteado

Silva, Tassalon Ferreira da
S578m Maximização da cobertura redundante em redes de sensores visuais sem fio para monitoramento de alvos / Tassalon Ferreira da Silva. - Feira de Santana, 2016.

88 f.: il.

Orientador: Daniel Gouveia Costa

Dissertação (Mestrado) – Universidade Estadual de Feira de Santana, Programa de Pós-graduação em Computação Aplicada, 2016.

1. Computação. 2. Redes de Sensores Visuais sem fio. 3. ECPGA. RCMA, EDPGA – algoritmos. I. Costa, Daniel Gouveia, orient. II. Universidade Estadual de Feira de Santana. III. Título.

CDU: 681.3

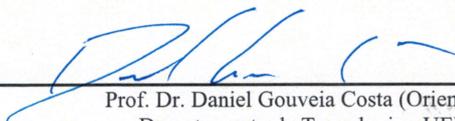
Tassalon Ferreira da Silva

**Maximização da Cobertura Redundante em Redes de Sensores
Visuais Sem Fio para Monitoramento de Alvos**

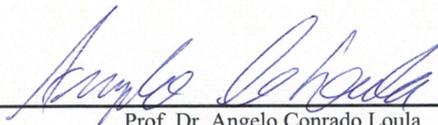
Dissertação apresentada à Universidade
Estadual de Feira de Santana como parte
dos requisitos para a obtenção do título
de Mestre em Computação Aplicada.

Feira de Santana, 18 de dezembro de 2015

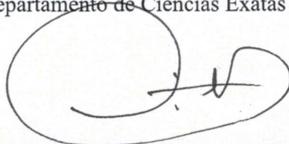
BANCA EXAMINADORA



Prof. Dr. Daniel Gouveia Costa (Orientador)
Departamento de Tecnologia - UEFS



Prof. Dr. Angelo Conrado Loula
Departamento de Ciências Exatas - UEFS



Prof. Dr. Cristian Duran-Faundez
Universidade del Bio-Bio (Chile)

Abstract

This work have proposed the adaptation and the development of algorithms for maximization of the redundant coverage of targets in wireless visual sensor networks. The most relevant concepts were described about the theory of visual sensors networks, development techniques of coverage algorithms and the details of the performed implementations. Besides, tests and comparisons were performed in simulation environments to demonstrate the obtained results, as well as to evaluate the performance of each developed algorithm of redundant coverage.

Keywords: Wireless Visual Sensors Network, Redundant Visual Coverage, Maximization of the Redundant Coverage

Resumo

Esse trabalho propõe a adaptação e o desenvolvimento de algoritmos para a maximização da cobertura visual redundante de alvos em redes de sensores visuais sem fio. Assim, serão apresentados os conceitos mais importantes relacionados a teoria de redes de sensores visuais, as técnicas de desenvolvimento de algoritmos de cobertura e os detalhes das implementações realizadas. Além disso, testes e comparações foram realizados em ambiente de simulação para demonstrar os resultados obtidos, bem como avaliar o desempenho de cada um dos algoritmos de cobertura redundante desenvolvidos.

Palavras-chave: Redes de Sensores Visuais Sem Fio, Cobertura Visual Redundante, Maximização da Cobertura Redundante.

Prefácio

Esta dissertação de mestrado foi submetida a Universidade Estadual de Feira de Santana (UEFS) como requisito parcial para obtenção do grau de Mestre em Computação Aplicada.

A dissertação foi desenvolvida dentro do Programa de Pós-Graduação em Computação Aplicada (PGCA) tendo como orientador o Dr. **Daniel Gouveia Costa**.

Esta pesquisa foi parcialmente financiada pela Fapesb.

Agradecimentos

Agradeço a Deus por tudo que Ele tem feito em minha vida e por mais essa vitória alcançada. Agradeço também aos meus pais, Maria Gilda Ferreira da Silva e José Abiderman Lopes da Silva, por todo o suporte e amor que me deram. Sou grato também a minha companheira Karine Souza de Almeida que me deu todo apoio que precisei nesses últimos dois anos. Agradeço ainda ao meu orientador Dr. Daniel Gouveia Costa por todo esforço e dedicação em me ajudar sempre que precisei. Além disso, agradeço ainda aos colegas de curso e aos amigos do NuCOPE, pelos momentos de discussão de ideias e de recreação, que me ajudaram nessa caminhada para alcançar o título de Mestre em Computação Aplicada.

Sumário

Abstract	i
Resumo	ii
Prefácio	iii
Agradecimentos	iv
Sumário	vi
Lista de Publicações	vii
Lista de Tabelas	viii
Lista de Figuras	x
Lista de Abreviações	xi
1 Introdução	1
1.1 Justificativa	3
1.2 Relevância	3
1.3 Motivação	4
1.4 Objetivos	4
1.4.1 Objetivos Específicos	4
2 Revisão Bibliográfica	6
2.1 Redes de Sensores Sem Fio	6
2.2 Redes de Sensores Visuais Sem Fio	11
2.3 Cobertura em Redes de Sensores Visuais Sem Fio	15
2.3.1 Monitoramento Visual	15
2.3.2 Otimização de Cobertura	21
2.3.3 Redundância Visual	25
3 Metodologia	29
3.1 Formalização do Problema	30

3.2	Técnicas e Procedimentos	32
3.3	Ferramentas	34
3.4	Cenários de Testes	35
4	Algoritmos Propostos	37
4.1	ECPGA: Um algoritmo Guloso Aprimorado	37
4.1.1	1ª Melhoria: Aumento do número de Regiões de Cobertura . . .	38
4.1.2	2ª Melhoria: Ampliando a Quantidade de Alvos Cobertos . . .	41
4.1.3	Testes e Comparações	44
4.2	RCMA: Algoritmo de Maximização da Cobertura Redundante	48
4.2.1	Testes e Comparações	54
4.3	EDPGA: Um Algoritmo Distribuído	58
4.3.1	Testes e Comparações	62
5	Conclusão	69
	Referências Bibliográficas	71

Lista de Publicações

- Artigos submetidos e Aceitos

Silva, T.F. e Costa, D.G. (2015), ECPGA: Um algoritmo para maximização da cobertura redundante em redes de sensores visuais sem fio, Simpósio Brasileiro de Automação Inteligente (SBAI), 6p.

Almeida, K.S. de, Silva, T.F. e Costa, D.G. (2015), Uma Ferramenta para Suporte ao Ensino do Protocolo AODV, Associação Brasileira de Educação de Engenharia (ABENGE), Revista de Ensino de Engenharia, Volume 34, Número 2, 12p.

- Artigos submetidos e em Avaliação

Silva, T.F. e Costa, D.G. (2015), Centralized Algorithms for Redundant Coverage Maximization in Wireless Visual Sensor Networks, Revista IEEE América Latina, 7p.

Lista de Tabelas

- 4.1 Resultados obtidos nos testes dos algoritmos CPGA e ECPGA. . . . 46
- 4.2 Resultados obtidos nos testes dos algoritmos centralizados e do EDPGA 64

Lista de Figuras

2.1	Visão geral de uma rede de sensores.	7
2.2	Exemplos de sensores comerciais usados em redes de sensores sem fio.	7
2.3	Algumas aplicações das redes de sensores sem fio.	8
2.4	Comparação entre a região de cobertura das redes de sensores sem fio (RSSF) e das redes de sensores visuais sem fio (RSVSF).	12
2.5	Exemplos de sensores visuais disponíveis no mercado.	13
2.6	Região de cobertura de um sensor visual [Munishwar et al. 2014].	15
2.7	Modelo 2D da região de cobertura de um sensor visual [Costa et al. 2014b].	16
2.8	Diferenças na região de cobertura entre os sensores escalares e os sensores visuais.	17
2.9	Comparação entre implantação aleatória e implantação determinística de sensores visuais.	18
2.10	Rede de sensores visuais sem fio com cobertura redundante.	19
2.11	Exemplo de camera PTZ.	19
2.12	Camera PTZ ajustando seu campo de visão (FoV) para cobrir os alvos.	20
2.13	Rede de sensores visuais com prioridade de alvos.	20
2.14	Aumento da região de cobertura em uma RSVSF [Tezcan e Wang 2008].	22
2.15	Funcionamento do algoritmo FoVIC para otimização da cobertura.	23
2.16	Execução do algoritmo guloso para cobrir todos os alvos 2 vezes ($k = 2$) [Fusco e Gupta 2009].	24
2.17	Algoritmo responsável em montar os <i>clusters</i> [Alaei e Barcelo-Ordinas 2009].	26
2.18	Simulação do algoritmo guloso para melhoria da redundância [Costa et al. 2014b].	27
2.19	Problema do algoritmo guloso na cobertura de alvos.	28
4.1	Diferentes setores para ajuste da orientação (α) do FoV dos sensores visuais [Costa et al. 2014b].	38
4.2	Novos setores para ajuste da orientação (α) do FoV dos sensores visuais.	39
4.3	Resultado da execução do algoritmo adaptado.	42
4.4	Comparação entre os algoritmos CPGA e ECPGA.	44
4.5	Comparação da redundância visual de alvos entre os algoritmos CPGA e ECPGA.	45

4.6	Aumento percentual na redundância visual da RSVSF quando utilizado o algoritmo adaptado.	47
4.7	Comparação da redundância visual da RSVSF entre o algoritmo CPGA e o ECPGA.	48
4.8	Criação de setores com intersecções para ajuste da orientação do FoV dos sensores visuais.	49
4.9	Resultados iniciais obtidos com a execução do algoritmo RCMA.	50
4.10	Resultados obtidos em cada passo da execução do algoritmo RCMA.	51
4.11	Resultados obtidos em cada passo do teste do algoritmo RCMA.	55
4.12	Resultado obtido em cada algoritmo.	56
4.13	Comparação entre os algoritmos CPGA, ECPGA e RCMA.	57
4.14	A configuração da RSVSF para cada algoritmo em um dos testes comparativos de desempenho.	58
4.15	Funcionamento do algoritmo EDPGA.	59
4.16	Comparação entre o algoritmo distribuído e os algoritmos centralizados.	63
4.17	Desempenho dos algoritmos desenvolvidos.	64
4.18	Resultado obtido por cada algoritmo variando o raio de alcance dos sensores visuais.	65
4.19	Resultado obtido por cada algoritmo variando o ângulo de visão dos sensores visuais.	66
4.20	Comparação entre os algoritmos quando os sensores visuais são posicionados de forma aleatória.	66
4.21	Configuração dos sensores visuais após a execução de cada algoritmo no teste de nº 10.	67

Lista de Abreviações

Abreviação	Descrição
RSSF	Redes de Sensores Sem Fio
WSN	Wireless Sensor Network
RSVSF	Redes de Sensores Visuais Sem Fio
WVSN	Wireless Visual Sensor Networks
FoV	Field of View
DoF	Depth of Field
AoV	Angle of View
PTZ	Pan-Tilt-Zoom
CPGA	Centralized Prioritized Greedy Algorithm
ECPGA	Enhanced Centralized Prioritized Greedy Algorithm
RCMA	Redundant Coverage Maximization Algorithm
EDPGA	Enhanced Distributed Prioritized Greedy Algorithm

Capítulo 1

Introdução

As redes de sensores ganharam a atenção do mundo nos últimos anos. Sensores sem fio são pequenos dispositivos com baixo custo e que possuem processamento e recursos computacionais limitados. Eles podem monitorar e coletar informações do ambiente, além de transmitir tais informações para um dispositivo remoto [Yick et al. 2008].

Em geral, os sensores sem fio são dispositivos equipados com uma ou mais unidades de monitoramento, processador, memória, suprimento de energia, rádio de comunicação e, em alguns casos, atuadores. Os sensores podem ser dos mais variados tipos (térmico, biológico, químico, etc), a depender da aplicação. As principais aplicações desses sensores são em monitoramento de ambientes e rastreamento de alvos, porém há um grande leque de possibilidades [Yick et al. 2008] [Akyildiz et al. 2002].

Estes sensores possuem uma interface de rede sem fio, permitindo que se comuniquem formando uma rede. Uma Rede de Sensores Sem Fio (RSSF), também conhecida como Wireless Sensor Network (WSN), consiste de vários nós sensores trabalhando em conjunto para um determinado propósito. O projeto de uma rede de sensores sem fio depende significativamente da aplicação e deve considerar fatores como o ambiente, o objetivo da aplicação, custo, hardware e restrições do sistema [Akyildiz et al. 2002].

Uma das principais questões nas redes de sensores é em relação ao consumo de energia. Apesar dos nós serem construídos visando baixo custo, de forma que se tornem potencialmente descartáveis, é fundamental que seu tempo de vida seja o mais longo possível para que possam cumprir as tarefas de uma aplicação. Portanto, vários estudos tentam minimizar o consumo de energia propondo protocolos de comunicação cada vez mais eficientes não só nas camadas física e de enlace como também nas camadas de rede e de aplicação [Baronti et al. 2007].

Outra alternativa para manter as redes de sensores ativas por mais tempo é através da redundância de monitoramento. A redundância é possível quando dois ou mais nós compartilham a mesma região de monitoramento, capturando informações

em um mesmo escopo. Instalar, em uma região de monitoramento, mais sensores do que o mínimo necessário para a aplicação permite que alguns nós (os nós redundantes) mantenham-se em estado inativo evitando, assim, o desperdício de energia. Durante a operação das redes de sensores, quando um nó ativo esgota sua energia, um nó redundante pode substituí-lo e manter a rede funcionando por mais tempo [Costa e Guedes 2010].

Quando nós em uma rede de sensores são equipados com câmeras é possível capturar, processar e transmitir informações visuais (imagens e vídeos), surgindo as redes de sensores visuais sem fio (RSVSF). Porém, sensores equipados com câmeras necessitam de maior poder de processamento, mais memória e consomem mais energia, gerando novos desafios a serem enfrentados [Yap e Yen 2014]. Assim, diversos trabalhos estão focados na melhoria da eficiência no consumo de energia, na segurança da transmissão, na exploração das características de codificação de vídeos e imagens e do fornecimento de diferentes níveis de qualidade de serviço (QoS) [Costa e Guedes 2010]. Porém, embora a confiabilidade seja altamente importante para as redes de sensores visuais, a disponibilidade será frequentemente desejada para aplicações críticas de monitoramento visual. E uma das questões chaves da disponibilidade é a redundância [Costa e Guedes 2014].

A redundância de sensores visuais pode ser considerada como a sobreposição do campo de visão de dois ou mais sensores visuais, o que pode indicar que estes sensores estão obtendo informações equivalentes [Costa et al. 2014a]. Entretanto, por se tratar de informações visuais, essa equivalência pode não ser aproveitada por determinadas aplicações. Isso se deve a orientação na qual os sensores visuais sobrepostos foram implantados em relação à região monitorada [Costa e Guedes 2014]. Além disso, é desejável que a região de cobertura das redes de sensores visuais alcance todos os alvos ou toda a região de interesse da aplicação de forma a não comprometer os resultados esperados [Yap e Yen 2014]. Para isso, diversas abordagens foram propostas na literatura visando otimizar a cobertura das redes de sensores visuais de forma que se alcance a cobertura total dos alvos ou da região de interesse com o menor número de sensores possíveis [Munishwar e Abu-Ghazaleh 2013a] [Ai e Abouzeid 2006] [Yen 2013].

Em muitos casos, a otimização da cobertura é possível quando os sensores visuais, ou apenas a sua câmera, são móveis e podem se deslocar ajustando seu campo de visão para abranger todos os alvos ou as áreas de interesse da aplicação. Existem ainda aplicações nas quais determinados alvos têm maior prioridade de monitoramento que outros. Nestas aplicações, o posicionamento do campo de visão dos sensores deve priorizar os alvos mais importantes, quando não for possível a cobertura de todos os alvos [Costa e Guedes 2014].

Com isso, muitos trabalhos foram desenvolvidos nos últimos anos visando a otimização da cobertura visual, em que a redundância é explorada para diminuir o número de nós ativos e, assim, reduzir o consumo de energia [Thangaraj e Chitra 2012] [Li e Zhang 2013] [Alaei e Barcelo-Ordinas 2009].

Porém, embora a otimização da cobertura seja uma importante questão de projeto, aplicações críticas demandam a maximização da cobertura redundante, em que múltiplos sensores se mantêm ativos quando estão monitorando o mesmo alvo ou região. Dessa forma, é possível obter melhorias na disponibilidade da rede, com rápida resposta a eventos críticos. Por fim, a ativação simultânea de sensores redundantes oferece múltiplas perspectivas dos alvos monitorados, aumentando a precisão dos resultados obtidos pela aplicação [Costa e Guedes 2014].

1.1 Justificativa

A utilização de sensores com capacidade para monitoramento multimídia permitiu o surgimento das redes de sensores visuais sem fio. Nesse tipo de rede, os nós podem ter mais recursos de processamento e memória uma vez que tratam dados como imagem e vídeo. Contudo, a natureza de transmissão desses dados e as restrições impostas pelas redes de sensores visuais criam um ambiente complexo que apresenta muitos desafios de projeto, implantação, operação e gerenciamento.

Dentro desse contexto, um dos principais problemas está relacionado com a melhoria do monitoramento. Assim, para manter a disponibilidade das redes de sensores visuais por mais tempo, é necessário desenvolver técnicas para o aumento da redundância do campo de visão dos sensores, permitindo que os nós principais possam ser substituídos pelos redundantes quando a sua energia se esgota. Paralelamente, a cobertura de todos os alvos, ou daqueles que apresentam maior prioridade, deve ser levada em consideração na implantação da rede para não comprometer os resultados da aplicação.

Ainda são poucos os trabalhos que abordam esse tema, por isso, desenvolver abordagens que possam conciliar a otimização da cobertura com a maximização da redundância, é de grande importância para contribuir para a melhoria da disponibilidade das redes de sensores visuais sem fio, sobretudo em aplicações críticas. Uma vez que isso é alcançado, é possível fazer com que as redes de sensores visuais sem fio tornem-se mais disponíveis e/ou objetivas em diversas aplicações de controle e monitoramento.

1.2 Relevância

A maioria das abordagens desenvolvidas até então a cerca da otimização da cobertura nas redes de sensores visuais estão voltadas em obter a maior cobertura com o menor número de sensores, sem levar em consideração o aumento da redundância visual da rede para melhorar a disponibilidade e obter diferentes perspectivas do alvo monitorado [Yen 2013] [Almalkawi et al. 2010] [Fusco e Gupta 2009]. Este trabalho propõe desenvolver adaptações das técnicas tradicionais de otimização de cobertura

em redes de sensores visuais para resolver um tipo diferente de problema, em que se deseja encontrar a melhor configuração quando todos os sensores da rede estão ativos. Com isso, o aumento da redundância visual permitirá que as aplicações tenham diferentes perspectivas dos alvos monitorados, aumentando a precisão dos dados coletados e possibilitando o surgimento de novas aplicações.

Para tanto, os sensores visuais, além de estarem posicionados de forma a fornecer uma cobertura total dos alvos, devem também estar dispostos de forma que haja a maior sobreposição de campos de visão entre os sensores para assim aumentar a redundância da rede. A prioridade dos alvos também será considerada no desenvolvimento das adaptações de forma que seja possível se adequar às diferentes aplicações das redes de sensores visuais sem fio.

1.3 Motivação

As redes de sensores visuais sem fio têm uma ampla gama de aplicações no mundo real, desde as mais simples, como o monitoramento de regiões de interesse, às mais complexas, como o rastreamento de alvos e reconhecimento de padrões. O desenvolvimento de novas abordagens que possam levar essas redes a se manterem por mais tempo em operação as tornam mais eficazes, confiáveis e populares, incentivando e justificando ainda mais o uso das redes de sensores visuais em uma quantidade cada vez maior de aplicações.

Portanto, este trabalho visa contribuir para a ciência e tecnologia ao ter como objetivo desenvolver algoritmos de cobertura visando o aumento da redundância visual das redes de sensores visuais que podem ser usadas em diversas aplicações reais. Além disso, a proposta desse trabalho se ajusta adequadamente ao programa de pós-graduação em computação aplicada ao abordar um tema que envolve conceitos de áreas diversas do conhecimento e utiliza de técnicas computacionais para sua solução.

1.4 Objetivos

Adaptar e desenvolver técnicas de otimização da cobertura visando a maximização da redundância visual em redes de sensores visuais sem fio.

1.4.1 Objetivos Específicos

- Apresentar as técnicas de otimização de cobertura adaptadas e desenvolvidas para aumentar a redundância nas redes de sensores visuais sem fio.
- Avaliar o desempenho dos algoritmos adaptados e desenvolvidos.

- Difundir os resultados obtidos através de dissertação de mestrado e elaboração de artigos científicos.

Capítulo 2

Revisão Bibliográfica

2.1 Redes de Sensores Sem Fio

Recentes avanços da tecnologia têm permitido o desenvolvimento de pequenos sensores de baixo custo, com suprimento de energia, processamento e recursos computacionais limitados. Eles podem medir e colher informações do ambiente além de transmitir tais informações para um *host* através de uma interface sem fio. O conjunto de diversos sensores como este, trabalhando unidos para uma determinada aplicação, compõe o que se chama hoje de redes de sensores sem fio (RSSF) ou wireless sensor networks (WSN) [Akyildiz et al. 2002].

Dos sensores disponíveis, existem os genéricos (com múltiplas finalidades) e os sensores *gateway*, também chamados de *sink*. Os genéricos realizam tarefas como a coleta de medidas de ambientes monitorados e comunicação entre sensores. Eles podem ser equipados com diversos dispositivos que podem medir várias grandezas físicas, como temperatura, umidade, pressão, velocidade, entre outros [Yick et al. 2008]. Já sensores *gateway* reúnem informações dos sensores genéricos e enviam para estações base. Eles têm maior capacidade de processamento, de alimentação e de alcance de transmissão [Borges et al. 2014]. Uma rede de sensores sem fio é tipicamente implementada com um conjunto de nós genéricos e de *gateway* como ilustrando na Figura 2.1.

Os sensores variam de preço, tamanho, capacidade de armazenamento e forma de comunicação. Geralmente um nó é um microcontrolador com transdutores acoplados que medem temperatura, pressão, posição, etc e uma fonte de alimentação, como baterias AA. Os transdutores se comunicam com o microcontrolador através de barramentos e protocolos de comunicação, como UART. A comunicação entre os sensores na maioria dos casos se dá por radiofrequência (RF) devido suas características mais apropriadas para as redes de sensores [Baronti et al. 2007]. A Figura 2.2 ilustra alguns sensores usados nas redes de sensores.

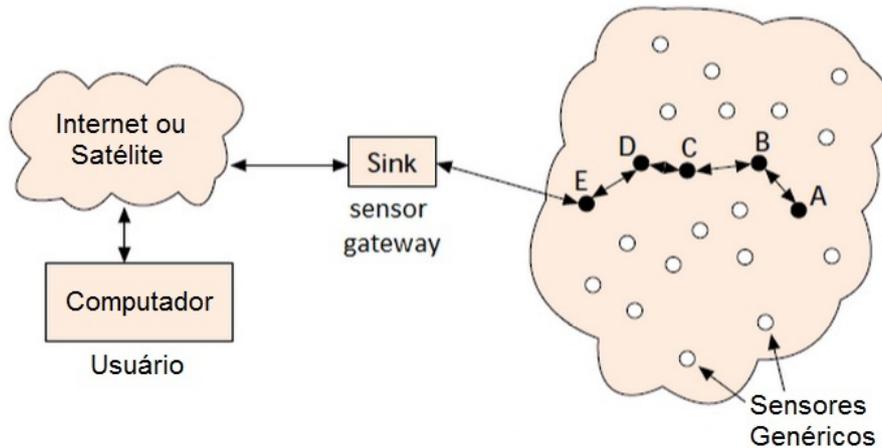


Figura 2.1: Visão geral de uma rede de sensores.

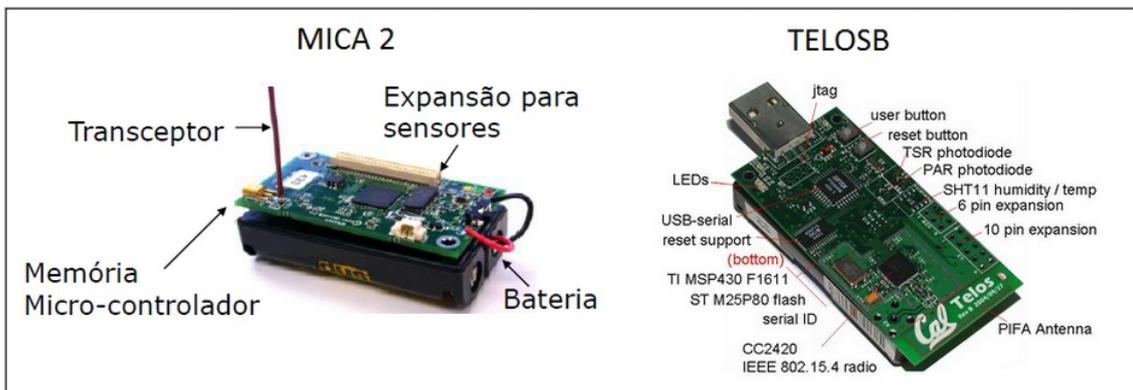


Figura 2.2: Exemplos de sensores comerciais usados em redes de sensores sem fio.

Diferente de outras redes, as redes de sensores sem fio são projetadas para aplicações específicas. Os sensores podem ser implantados na terra, no subsolo, na água ou até mesmo no espaço. Para isso, existem 5 tipos de redes de sensores sem fio: terrestre, subterrâneo, subaquático, multimídia e móvel [Yick et al. 2008]. As principais aplicações das RSSF são para monitoramento e rastreamento. Elas podem ser usadas, por exemplo, para monitorar condições climáticas de determinada região ou observar condições de vida de determinados animais ameaçados de extinção [Borges et al. 2014]. A Figura 2.3 apresenta um resumo das diversas aplicações das redes de sensores atualmente.

Para as aplicações com redes de sensores sem fio funcionarem, são necessárias várias tarefas que podem ser classificados em 3 principais grupos: sistemas, serviços e protocolos de comunicação. Cada sensor tem um sistema individual, e para suportar as diferentes aplicações é necessário o desenvolvimento de plataformas, sistemas operacionais e esquemas de armazenamento para os nós de sensores. Os serviços provêm

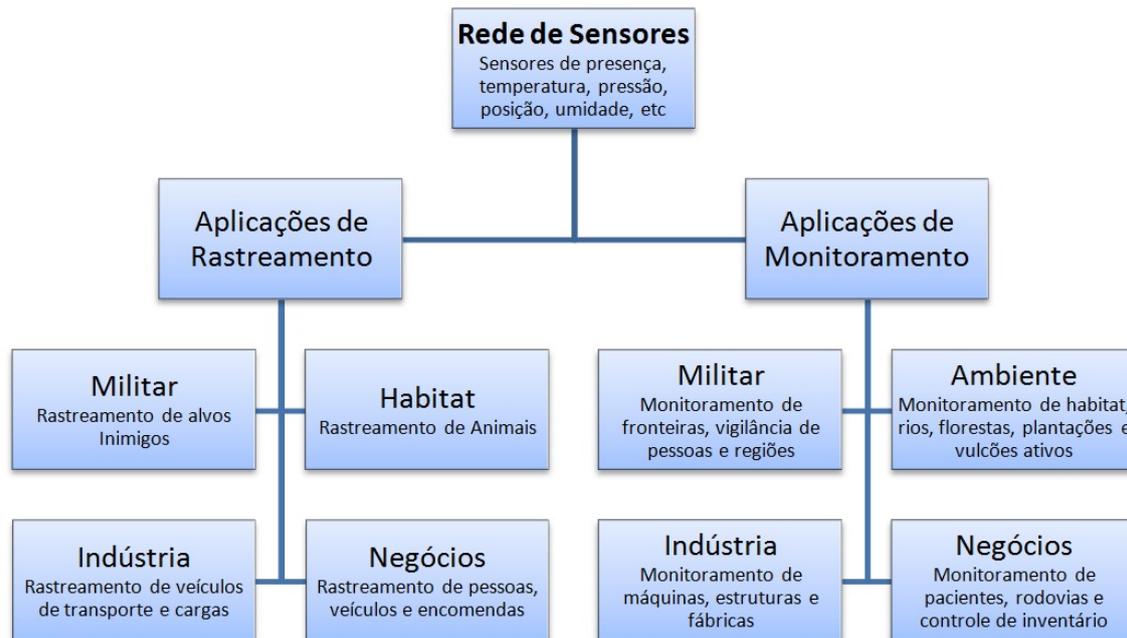


Figura 2.3: Algumas aplicações das redes de sensores sem fio.

melhorias para aplicação, como o aumento da eficiência da rede e do desempenho do sistema. Já os protocolos de comunicação são responsáveis pela troca de informações entre sensores e entre a aplicação [Yick et al. 2008].

Como os sensores possuem limitados recursos de energia, processamento e armazenamento, se torna necessário o desenvolvimento de novos protocolos de comunicação e serviços de gerenciamento para atender esses requisitos, uma vez que protocolos tradicionais de redes não atendem às necessidades das redes de sensores sem fio, pois não foram projetados para esses requisitos de recursos limitados. Além disso, é importante que os nós de sensores sejam capazes de se auto-organizar. Ou seja, os sensores devem ter a capacidade de se unirem em uma rede, gerenciando-se de forma eficiente para fazer melhor uso dos limitados recursos disponíveis [Yick et al. 2008].

O desenvolvimento dos protocolos de comunicação nas redes de sensores sem fio é dividido em 5 camadas que se intercomunicam fornecendo o serviço de comunicação fim a fim. As camadas são: camada de Aplicação, Transporte, Rede, Enlace de Dados e Física [Akyildiz et al. 2002]. A implementação de diversos protocolos em diferentes camadas da pilha de protocolos pode afetar a eficiência do sistema, o consumo de energia e o atraso fim a fim. Dessa forma, é importante otimizar o desenvolvimento desses protocolos para aumentar a eficiência da comunicação e economizar o uso de energia [Yick et al. 2008].

Há diversos protocolos disponíveis para as redes de sensores sem fio, sendo que alguns se destacam mais devido suas características serem mais adequadas para as RSSF.

Dentre esses, tem-se o protocolo ZigBee que busca levar confiabilidade, baixo custo e baixo consumo para as redes sem fio [Gungor e Hancke 2009]. Essa tecnologia já é um padrão mundialmente conhecido e embarcado em diversos produtos e aplicações das mais diversas áreas, como indústria, comércio e governo [Yick et al. 2008]. O ZigBee opera na camada de rede e oferece um framework para o desenvolvimento de aplicações na camada de aplicação e ainda fornece segurança, provendo serviços de criptografia, autenticação e integridade. Ele foi construído sobre o protocolo IEEE 802.15.4 que opera nas camadas físicas e enlace [Baronti et al. 2007].

O protocolo 802.15.4 foi criado para redes sem fio pessoais (WPAN) tendo como características curta distância de operação, baixo custo de implantação e reduzido consumo de energia, além de simplicidade de manutenção e flexibilidade. Funciona em frequências baixas (868/915 MHz) e altas (2.4 GHz), cada uma com velocidades distintas. A mais rápida opera a 250 kbps na frequência mais alta, de 2.4 GHz [Baronti et al. 2007].

Os protocolos ZigBee e o IEEE 802.15.4 são os mais comumente usados nas redes de sensores sem fio, porém há outros protocolos que buscam atender requisitos mais específicos de cada aplicação, como o 6LoWPAN que oferece suporte ao IPv6 e é ideal para aplicações de RSSF que necessitem de comunicação com a Internet com baixo tráfego de dados [Gungor e Hancke 2009]. Quando se necessita de alta velocidade de comunicação para aplicações multimídia e de tempo real, pode-se utilizar o protocolo IEEE 802.15.3 que atinge velocidade de até 55 Mbps e oferece garantia de qualidade de serviço (QoS), porém apresenta maior consumo de energia [Yick et al. 2008].

Um dos fatores mais críticos das redes de sensores sem fio é o consumo de energia, que depende de diversas variáveis. A distância entre cada nó da rede de sensores pode aumentar a cobertura da rede, porém exige mais consumo de energia para manter os nós se comunicando. Além disso, colisões e perdas de pacotes, espera ociosa pelo recebimento de pacotes entre outras especificidades das comunicações sem fio é uma das principais causas do desperdício de energia das RSSF [Boonsongsrikul et al. 2013]. A taxa de comunicação, o armazenamento e o processamento dos dados coletados também influenciam no consumo de energia. Geralmente os sensores são alimentados por baterias acopladas, mas também podem adquirir energia de outras fontes como energia solar ou células de combustível ou ainda de fontes móveis de energia através do uso de robôs para recarga [Yick et al. 2008].

Os nós são construídos para terem baixo custo de forma que se tornem descartáveis, porém é fundamental que seu tempo de vida seja o mais longo possível para que possa cumprir as tarefas de uma aplicação. Os sensores atuais já permitem que aplicações liguem e desliguem componentes de *hardware* em momentos ociosos para evitar o consumo desnecessário de energia. Entretanto, vários estudos ainda buscam reduzir o consumo de energia fazendo protocolos de comunicação cada vez mais eficientes em todas as camadas da pilha de protocolos de comunicação [Baronti et al. 2007]. Estudos mais recentes propõem o desenvolvimento de protocolos *cross-layer*, que tentam otimizar a estrutura em camadas e construir protocolos mais independentes

que possam interagir diretamente com protocolos de diversas camadas visando reduzir o consumo de energia e melhorar a eficiência das RSSF [Gungor e Hancke 2009].

Além do consumo de energia, há outros fatores também importantes para as redes de sensores sem fio. Questões como a localização dos sensores implantados, roteamento de pacotes, gerenciamento de dados, cobertura, disponibilidade e segurança da rede são temas fundamentais para o bom funcionamento das aplicações que utilizam as RSSF [Baronti et al. 2007].

Em muitas aplicações, os nós das redes de sensores são implantados aleatoriamente no ambiente a ser monitorado. Saber a localização desses nós é importante tanto para os protocolos de roteamento fornecer rotas mais eficientes quanto para identificar o local de origem dos dados coletados, necessário em determinadas aplicações. Para isso, os sensores podem ser equipados com sistema de posicionamento global (GPS - Global Positioning System) para determinar a localização onde foram implantados. Porém, nem sempre é possível colocar em todos os nós um GPS para determinar sua localização, devido o ambiente onde é implantado ou as limitações de recursos do nó, como a quantidade de energia disponível. Para amenizar esses problemas, diversas técnicas são desenvolvidas de forma a melhorar a determinação da localização dos nós, mesmo que haja uma queda na precisão das informações. Alguns exemplos de técnicas de localização são o RSSI, TDoA e AoA [Baronti et al. 2007]. Em resumo, essas técnicas buscam extrair informações sobre a localização dos nós explorando as características da comunicação sem fio (como o tempo de resposta e a perda de propagação durante a transmissão de dados).

O objetivo final de uma rede de sensores é fornecer aos usuários dados relevantes obtidos pelos nós. Para isso, os usuários interagem com a aplicação (por meio de um computador e uma interface) determinando quais dados são importantes e devem ser analisados. Os sensores então monitoram o ambiente onde foram implantados e enviam através da rede de sensores sem fio as informações definidas pelo usuário. Porém, durante a comunicação entre os nós da rede de sensores, é possível que ocorra perda de conexão e falhas de sensores na rede prejudicando a precisão dos dados coletados. Para tratar esses problemas, uma atenção especial deve ser dada a camada de aplicação que é responsável pelo gerenciamento e coleta dos dados. Além disso, existem diversas abordagens para a coleta desse dados, e o uso de cada uma delas deve está de acordo com os requisitos da aplicação [Baronti et al. 2007].

Quando uma aplicação necessita de dados mais confiáveis, ou o ambiente de operação das RSSF possui muitos obstáculos e interferências, torna-se necessário o uso de muitos sensores sob a mesma região para prover um alto grau de cobertura do ambiente e assim coletar dados mais precisos e consistentes. Porém, nem sempre é viável para a aplicação o uso de muitos sensores, por questões de custo [Garcia-Hernandez et al. 2007]. Com isso, diversas pesquisas buscam criar técnicas de implantação de sensores para maximizar a cobertura com a menor quantidade de nós e assim obter resultados satisfatórios para a aplicação e com custo reduzido [Munishwar e Abu-Ghazaleh 2013a] [Ai e Abouzeid 2006] [Yen 2013].

Enquanto isso, há outras aplicações onde a disponibilidade da rede é um fator crucial e deve ser priorizada. Aplicações críticas que monitoram ambientes perigosos e que podem causar riscos à vida precisam manter a rede em funcionamento em qualquer condição. Uma das formas de manter as redes de sensores ativas e funcionando por mais tempo é aumentar a redundância de nós implantados. Dessa forma, a redundância pode ser usada para aumentar o nível de disponibilidade da rede permitindo a substituição dos nós que falham por nós redundantes capazes de dar continuidade a operação da rede [Costa e Guedes 2014].

Por fim, uma rede de sensores sem fio é vulnerável a ameaças de segurança. Um intruso pode comprometer um sensor, alterar a integridade dos dados, espionar mensagens, injetar mensagens falsas e desperdiçar recursos da rede. Diferente de redes cabeadas, as redes de sensores enviam suas mensagens em *broadcast* pelo meio sem fio podendo mais facilmente serem interceptadas. Assim, questões de segurança também devem ser tratadas nas RSSF. Porém, projetar protocolos seguros para RSSF tem limitações em termos de processamento, comunicação, armazenamento e, conseqüentemente, energia. Atualmente, há mecanismos de segurança para as diversas camadas da pilha de protocolos e, em geral, elas buscam oferecer confidencialidade, autenticidade e disponibilidade de dados [Yick et al. 2008].

Mais recentemente, redes de sensores multimídias têm sido propostas para monitoramento e rastreamento de alvos na forma multimídia (coletando vídeo, áudio e/ou imagem). Esse tipo de RSSF consiste em um número de sensores de baixo custo equipados com câmeras e microfones [Akyildiz et al. 2007]. Os desafios dessas redes de sensores multimídias inclui a alta largura de banda necessária para transmissão de dados multimídia, a redução do alto consumo de energia, o fornecimento de qualidade de serviço e a melhoria de técnicas de processamento e compressão de dados [Yick et al. 2008].

2.2 Redes de Sensores Visuais Sem Fio

Devido ao grande avanço da tecnologia é possível equipar sensores com câmeras e permitir a captura de dados visuais (tanto de imagens como de vídeos) da região de monitoramento. Dessa forma, informações mais detalhadas da área monitorada podem ser capturadas, aumentando o escopo de aplicações possíveis para as redes de sensores. Esse tipo de rede de sensores equipados com câmeras é conhecido como redes de sensores visuais sem fio (RSVSF) ou *wireless visual sensor networks* (WWSN) [Akyildiz et al. 2007].

As RSVSF se diferenciam das redes de sensores sem fio convencionais em diversos aspectos. O consumo de largura de banda das RSVSF é relativamente maior uma vez que o tamanho dos dados visuais transmitidos é superior ao dos dados escalares, como os obtidos nas redes de sensores tradicionais. Isso exige melhorias nos protocolos de roteamento e no desenvolvimento de mecanismos para oferecer qualidade

de serviço visando controlar os atrasos, a latência e a perda de pacotes durante a transmissão dos dados [Yap e Yen 2014].

Com isso, para obter maior eficiência nas RSVSF, muitas pesquisas exploram o conceito de projeto *cross-layer* para obter maiores taxas de transferência de dados, menor consumo de energia e manter aceitável a qualidade da aplicação. Protocolos *cross-layer* rompem o conceito de camadas modulares, reduzindo a sobrecarga da rede e otimizando seu desempenho [Costa e Guedes 2011]. O projeto de protocolos *cross-layer* entre as camadas de rede e de transporte, por exemplo, possibilita otimizar a seleção das melhores rotas, em redes com múltiplos caminhos, para garantir qualidade de serviço (QoS) e entrega confiável de dados nas redes de sensores visuais sem fio [Almalkawi et al. 2010].

Outro aspecto que diferencia as redes de sensores visuais sem fio das RSSF está relacionado a região de cobertura dos sensores. Nos sensores visuais a região de cobertura se limita ao campo de visão das câmeras acopladas, diferentemente das RSSF onde, geralmente, a cobertura é omnidirecional. Esse campo de visão é conhecido como FoV (Field of View) e os dados visuais capturados pelo sensor depende da direção e do ângulo de posicionamento da câmera [Yap e Yen 2014]. A Figura 2.4 ilustra a diferença da região de cobertura dos dois tipos de redes.

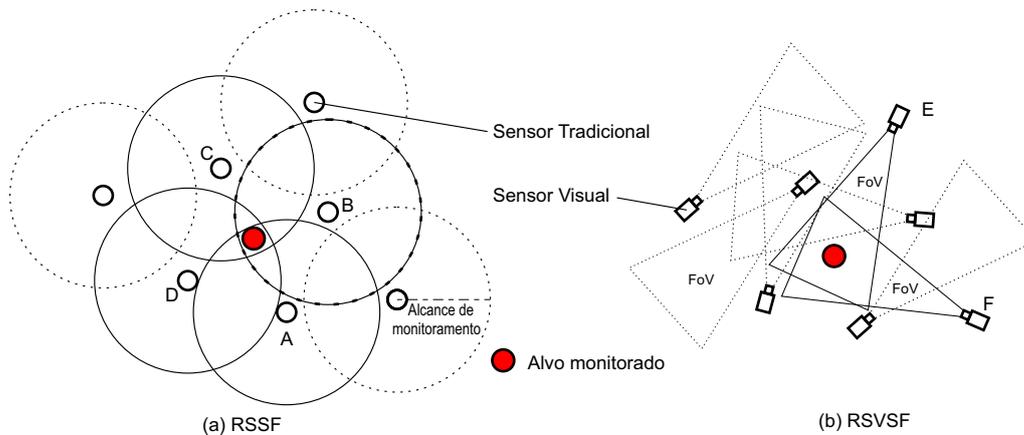


Figura 2.4: Comparação entre a região de cobertura das redes de sensores sem fio (RSSF) e das redes de sensores visuais sem fio (RSVSF).

A Figura 2.4(a) indica que o alvo monitorado está dentro da região de alcance dos sensores A, B, C e D de forma que estes podem capturar informações a cerca do objeto de interesse da aplicação, enquanto que a Figura 2.4(b) mostra que apenas os sensores visuais E e F podem capturar informações do alvo devido a direção do FoV destes sensores. Além disso, as informações visuais capturadas pelos sensores da RSVSF são diferentes, apesar de serem do mesmo alvo, uma vez que os sensores estão posicionados em locais distintos e a direção do campo de visão das câmeras são diferentes. A depender da aplicação, coletar dados visuais do alvo monitorado

em diferentes ângulos possibilita, além de maior disponibilidade para a rede, maior precisão dos resultados obtidos pela aplicação [Costa e Guedes 2014].

Entretanto, nas redes de sensores visuais sem fio, os nós acoplados com câmeras têm seu campo de visão (FoV) limitado. Esse campo de visão é influenciado por vários fatores como, por exemplo, qualidade da câmera, resolução, ângulo de visão, zoom e etc. Quanto maior for o ângulo de visão e o raio de alcance visual mais dados visuais a rede de sensores poderá obter, conseqüentemente o consumo de energia e o custo desse *hardware* será mais alto. O raio de comunicação entre os sensores visuais permanece omnidirecional e assim como ocorre nas redes de sensores sem fio tradicionais, as imagens coletadas pelos sensores visuais são enviadas para o *sink* através da rede e por fim transmitidas a um dispositivo final [Costa e Guedes 2011]. A Figura 2.5 apresenta alguns exemplos de sensores visuais disponíveis no mercado.



Figura 2.5: Exemplos de sensores visuais disponíveis no mercado.

A partir da Figura 2.5 é possível perceber que há uma variedade de sensores visuais e cada um apresenta características próprias de resolução de imagem, poder de processamento, capacidade de armazenamento, taxa de transmissão e consumo de energia [Almalkawi et al. 2010]. Dentre os sensores mais simples podemos citar o Cyclops com processador de 8 bits, memória RAM de 64 KBytes, câmera com resolução de 128x128 *pixels* além de usar o padrão ZigBee de comunicação sem fio atingindo taxas de até 250 Kbps e consumo máximo de energia estimado em 110 *mW* [Rahimi et al. 2005]. Já o sensor visual CITRIC [Chen et al. 2008], com desempenho mais robusto, é composto de um processador de 32 bits, memória RAM de 64 MBytes, equipado com uma câmera com resolução máxima de 1280x1024 *pixels* e operando sobre o padrão ZigBee com um consumo máximo de 1.000 *mW*.

Apesar da variedade disponível de *hardware* existente para compor redes de sensores visuais, o que irá definir quais dispositivos devem ser escolhidos é a aplicação na qual a rede será usada. Aplicações de reconhecimento facial, por exemplo, exigem câmeras de melhor qualidade de captura, com maior resolução de imagens, porém demandam maior consumo de energia e são mais caras. Entretanto, aplicações que realizam o monitoramento de objetos, mas não necessitam de alta qualidade de

imagens, podem usar *hardwares* mais simples com qualidade de captura de imagens inferior e que consomem menos energia e são mais baratos [Yap e Yen 2014].

Com o desenvolvimento das redes de sensores visuais sem fio surgiram novas aplicações que necessitam de coleta de imagens e vídeos. Por exemplo, vigilância de ambientes, monitoramento de desastres, assistência automatizada para idosos e deficientes, monitoramento de tráfego, controle de processos industriais e serviços de localização são algumas das aplicações que podem ser beneficiadas com a utilização de sensores visuais. Tais aplicações não são adequadas, ou até mesmo possíveis, apenas com o uso das redes de sensores sem fio tradicionais que realizam a coleta de dados escalares (como temperatura, umidade e pressão). Com isso, um novo campo de pesquisa surgiu com novos desafios a serem explorados [Costa e Guedes 2011].

Um desses desafios é o desenvolvimento de técnicas para a otimização da cobertura dos sensores visuais, uma vez que diversos fatores podem influenciar na cobertura dos alvos monitorados nas RSVSF. Assim, vários estudos propõem diferentes técnicas para a melhoria da cobertura de sensores visuais levando em consideração características como a limitação do FoV dos sensores visuais, a forma na qual os sensores são implantados (aleatório ou determinístico) e a mobilidade do FoV dos sensores visuais [Costa e Guedes 2010]. Além disso, uma demanda pouco explorada ainda é a maximização da cobertura redundante para aumentar a disponibilidade das RSVSF e obter diferentes perspectivas do alvo monitorado [Costa e Guedes 2014].

Outros desafios estão relacionados ao desenvolvimento de técnicas de codificação de vídeo e imagem para reduzir o tráfego de dados na rede, criação de métodos eficientes para o controle de congestionamento evitando o desperdício de energia devido a retransmissão de pacotes, além do uso de novos protocolos nas camadas física e de enlace para obter maiores taxas de comunicação, menor sobrecarga de controle, aumentar a confiabilidade da transmissão e garantir certos níveis de qualidade de serviço (QoS) [Almalkawi et al. 2010].

Por fim, muitas aplicações das redes de sensores visuais sem fio apresentam requisitos adicionais de privacidade e segurança, como aplicações militares, aplicações relacionadas a cuidados médicos, entre outros sistemas críticos de monitoramento visual [Almalkawi et al. 2010]. Nesse contexto, um dos mais complexos desafios da segurança é proteger a rede de ataques de negação de serviço (DoS) que podem ser realizados de inúmeras formas e em qualquer camada dos protocolos de comunicação. Assim, para garantir a autenticidade, integridade e confidencialidade dos dados transmitidos e também prevenir as RSVSF de ataques de negação de serviço (DoS), é necessário definir quais serão os requisitos de segurança da aplicação de forma a adotar medidas para a proteção da rede, porém, sem comprometer a sua eficiência [Guerrero-Zapata et al. 2010].

2.3 Cobertura em Redes de Sensores Visuais Sem Fio

2.3.1 Monitoramento Visual

A cobertura de um sensor visual é definida como a área efetiva na qual sua câmera consegue capturar imagens de objetos de forma nítida, sendo esta área também chamada de campo de visão (FoV - Field of View). Outra característica dos sensores visuais é a profundidade do campo de visão (DoF - Depth of Field), que é a distância entre o ponto mais perto do sensor visual " R_{min} " e o mais longe " R_{max} " que a câmera pode de fato visualizar um objeto com nitidez. Essa profundidade do campo de visão das câmeras, ou raio de alcance, é limitado devido às restrições de resolução e de distorção das lentes [Costa 2013]. Além disso, as câmeras também têm um limitado Ângulo de visão (AoV - angle of View), que é o ângulo máximo, a partir de determinada orientação, que a lente da câmera consegue observar um objeto. Em muitas câmeras esse ângulo de visão é definida tanto no sentido horizontal " θ_h " quanto vertical " θ_v " [Munishwar et al. 2014]. Todas essas características são importantes para a determinação do real campo de visão (FoV) dos sensores visuais. A Figura 2.6 ilustra a região de cobertura de uma câmera e as características envolvidas.

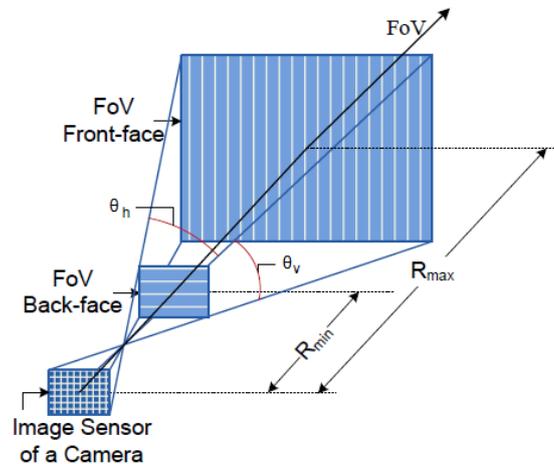


Figura 2.6: Região de cobertura de um sensor visual [Munishwar et al. 2014].

Para simplificar a modelagem matemática dos algoritmos de cobertura desenvolvidos, geralmente, o campo de visão (FoV) é representado em 2D, como mostra a Figura 2.7(a). Podendo ainda ser aproximada no formato de um triângulo isósceles composto de três vértices, A, B, e C, como pode ser visto na Figura 2.7(b), sendo o vértice A a posição atual do sensor visual. Assim, o ângulo de visão (AoV) considerado passa a ser apenas o de sentido horizontal e é representado por " θ ". Na Figura 2.7(b), também é apresentada a equação para o cálculo da área do FoV do sensor visual quando a região de cobertura é simplificada por um triângulo isósceles [Costa et al. 2014b]. Além disso, em geral, as câmeras não conseguem visualizar

de forma nítida objetos que estão muito próximos, porém, para simplificar os modelos matemáticos é comum considerar a profundidade do campo de visão (DoF) iniciando da posição do sensor visual (vértice A) e indo até o alcance máximo de reconhecimento de objetos. Assim, por questões de simplificação, o DoF é representado apenas pelo raio "R", como mostra a Figura 2.7(b).

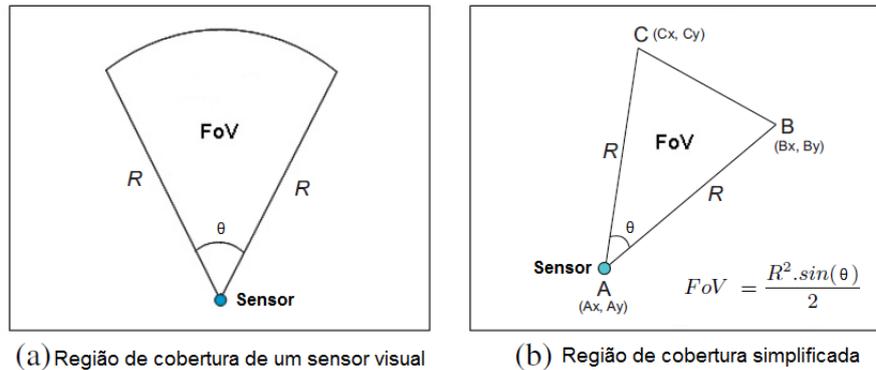


Figura 2.7: Modelo 2D da região de cobertura de um sensor visual [Costa et al. 2014b].

A cobertura é um problema crítico para as redes de sensores visuais sem fio já que as câmeras, geralmente, não oferecem cobertura omnidirecional, como é o caso dos sensores escalares tradicionais das RSSF. Porém, a região de cobertura das RSVSF, mesmo que limitada a uma direção específica, tem a capacidade de capturar dados visuais (vídeos e imagens) com maior raio de alcance que as RSSF, especialmente quando não há obstáculos (em linha de visada) [Almalkawi et al. 2010]. A Figura 2.8 ilustra essa diferença que há na região de cobertura entre os sensores escalares e o sensores visuais.

Na Figura 2.8, apesar dos sensores tradicionais (A, B e C) terem um ângulo de cobertura de 360° em torno de si, o raio de alcance deles é insuficiente para monitorar o alvo. Enquanto isso, o sensor visual E, com ângulo de cobertura limitado, consegue cobrir o alvo devido ao seu maior raio de alcance. Porém, é importante ressaltar que tanto o raio de alcance quanto o ângulo de cobertura dos sensores visuais são determinados pelo tipo de sensor usado na aplicação, o que pode variar a depender da qualidade dos sensores produzidos pelo fabricante. Entretanto, na maioria dos casos, os sensores visuais conseguem monitorar alvos a distâncias maiores do que os sensores escalares tradicionais [Almalkawi et al. 2010].

A Figura 2.8 também apresenta um dos grandes problemas enfrentados nas RSVSF que é a redução da cobertura dos sensores visuais quando há obstáculos no ambiente da aplicação. O fato dos sensores visuais terem um campo de visão (FoV) limitado pode influenciar diretamente na qualidade da aplicação uma vez que os dados coletados dependem basicamente do posicionamento dessas câmeras na região implantada. Ou seja, um posicionamento inadequado das câmeras poderá fazer com que alvos ou

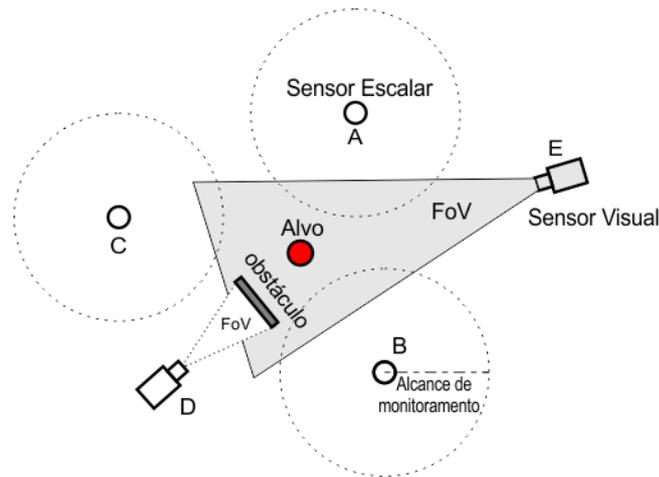


Figura 2.8: Diferenças na região de cobertura entre os sensores escalares e os sensores visuais.

regiões de interesse da aplicação não sejam cobertos, prejudicando o objetivo final da RSVSF [Costa e Guedes 2010].

Portanto, a forma como os nós são implantados é de grande importância para um melhor posicionamento do FoV dos sensores visuais. As redes de sensores visuais podem ser implantadas de forma determinística ou aleatória. Na implantação determinística os nós são colocados de maneira pré-programada. Dessa forma é possível maximizar a cobertura da rede com um mínimo número de sensores, reduzindo o custo final da RSVSF. Por outro lado, na implantação aleatória os sensores são dispersos na região monitorada sem um prévio conhecimento do ambiente. Isso acontece geralmente quando os sensores são lançados de um avião sobre uma região hostil ou de difícil acesso [Yap e Yen 2014]. A Figura 2.9 faz uma comparação entre as diferentes formas de implantação dos sensores visuais. Além disso, quando vários sensores com câmeras de baixa qualidade e baratos são implantados de forma aleatória sobre uma região de monitoramento, é possível obter maior disponibilidade e reduzir o consumo de energia. Tal opção é, muitas vezes, mais viável do que trabalhar com poucos nós (menor disponibilidade), com câmera de melhor qualidade (mais caro e com maior consumo de energia), implantados de forma determinística [Costa e Guedes 2010].

A Figura 2.9(a) apresenta sete sensores visuais implantados de forma aleatória sobre uma região de monitoramento com três alvos de interesse para a aplicação. Destes três alvos, apenas dois são cobertos pelo FoV dos sensores visuais o que afeta diretamente a qualidade dos resultados da aplicação. Percebe-se, então, que a implantação aleatória, apesar de inevitável em alguns casos, pode prejudicar a aplicação ao não conseguir cobrir todos os alvos de interesse. Enquanto isso, a implantação determinística tem o potencial de otimizar a cobertura da RSVSF de forma a alcançar todos os alvos de interesse da aplicação, como mostra a Figura 2.9(b). Além disso, percebe-se nessa figura que os sensores visuais A, B, C e D não são necessários para

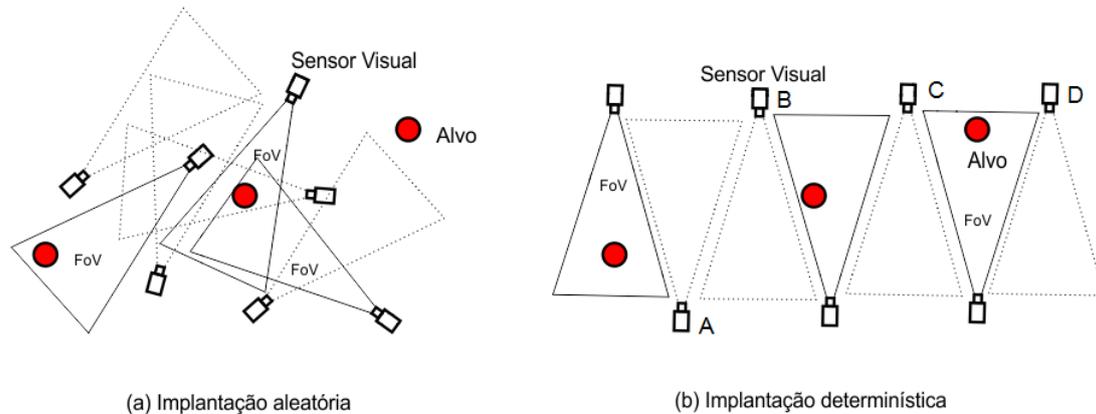


Figura 2.9: Comparação entre implantação aleatória e implantação determinística de sensores visuais.

garantir a cobertura dos três alvos contidos nesse ambiente. Assim, eles podem ser removidos para reduzir o custo da aplicação ou podem ser mantidos para ampliar a região de cobertura da RSVSF e monitorar outros alvos que possam surgir nesse ambiente.

Além da cobertura visual, uma outra preocupação nas redes de sensores visuais sem fio é com relação ao consumo de energia, que é ainda mais crítico que nas RSSF. Por isso, existem diversos algoritmos com várias abordagens para preservar a energia e manter a cobertura e conectividade da rede. A maioria se preocupa em como manter desligados os nós redundantes de forma a reduzir seu consumo de energia, porém sem fazer com que estes nós percam a conectividade com a rede para que possam identificar quando devem ser reativados e usados em substituição a outros nós que perderam sua energia ou sofreram problemas [Costa e Guedes 2010]. Mas, para isso, é necessário implantar na região monitorada uma quantidade excessiva de sensores de forma a aumentar a redundância visual e provocar a sobreposição do campo de visão (FoV), como mostra a Figura 2.10(a). É através dessa sobreposição que os nós redundantes poderão ser desligados preservando a sua energia. Quando um dos nós ativos estiver com sua fonte de energia esgotada, o nó redundante entra em ação aumentando assim o tempo de vida da rede e mantendo a sua cobertura, como ilustra a Figura 2.10(b).

Muitos trabalhos também exploram os recursos de mobilidade usando nós móveis capazes de alterar a região de monitoramento e maximizar a área de cobertura. Aliado a isso, é possível usar baterias de energia solar recarregáveis para aumentar a vida útil dos nós móveis e conseqüentemente da rede. Há também sensores estáticos com câmeras ajustáveis do tipo PTZ (*Pan-Tilt-Zoom*) que podem rotacionar sobre seu eixo horizontal e vertical alterando o seu campo de visão (FoV) para cobrir os alvos de interesse da aplicação, como mostra a Figura 2.11 [Munishwar et al. 2014]. Outra possibilidade é trabalhar com RSVSF heterogêneas visando a otimização dos

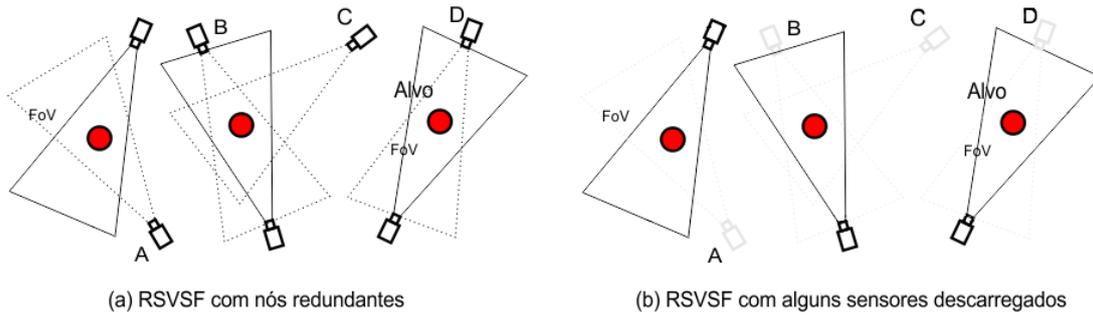


Figura 2.10: Rede de sensores visuais sem fio com cobertura redundante.

recursos, através do uso de diferentes modelos de sensores visuais [Costa 2013].

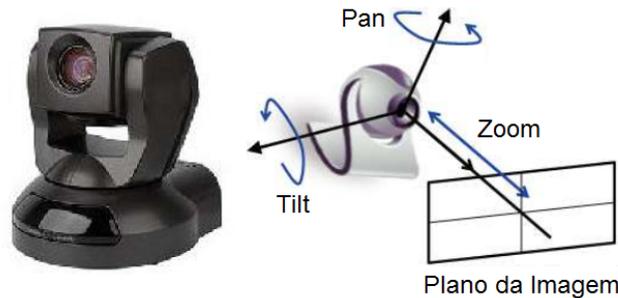


Figura 2.11: Exemplo de camera PTZ.

Uma câmera PTZ (*Pan-Tilt-Zoom*), como ilustrado na Figura 2.11, apresenta três propriedades básicas: *Pan*, responsável pelo movimento horizontal da câmera; *Tilt*, que torna possível o movimento vertical; e *Zoom*, que permite alterar a profundidade do campo de visão [Hatanaka et al. 2013]. Dessa forma, as câmeras do tipo PTZ permitem maiores possibilidades de cobertura para as aplicação das RSVSF. Porém, a maioria dos estudos propõe técnicas de melhoria da cobertura e da redundância visual considerando apenas a característica *Pan* desses sensores visuais, o que permite reduzir o problema para duas dimensões [Tezcan e Wang 2008]. Isso acontece devido a complexidade que há em se trabalhar em três dimensões, caso fossem consideradas as demais características das câmeras PTZ [Munishwar e Abu-Ghazaleh 2013b]. A Figura 2.12 ilustra a vantagem do uso de câmeras PTZ em uma rede de sensores visuais sem fio para melhoria da cobertura.

Quando câmeras PTZ são utilizadas nas redes de sensores visuais sem fio, como ilustrado na Figura 2.12, os nós podem ser implantados de forma aleatória (Figura 2.12(a)) e usar um algoritmo de otimização de cobertura para determinar a melhor posição das câmeras visando cobrir todos os alvos da aplicação (Figura 2.12(b)) [Munishwar et al. 2014]. Isso permite que aplicações em ambientes de alto risco, onde a implantação determinística não é possível, possam ter meca-

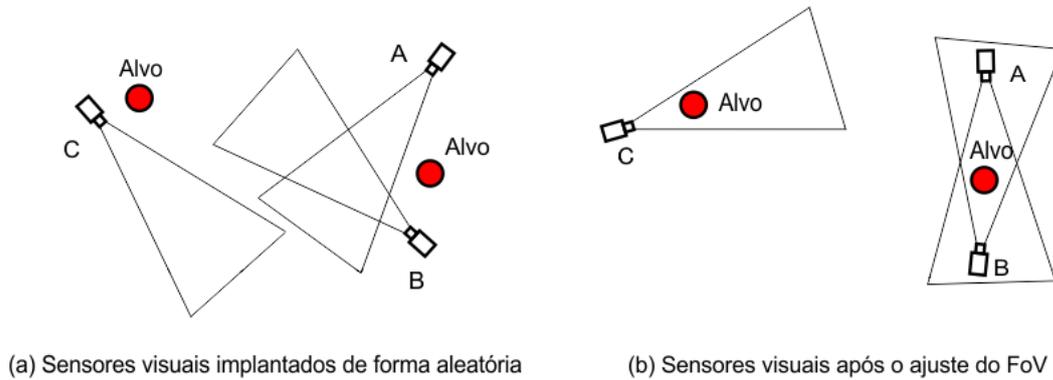


Figura 2.12: Camera PTZ ajustando seu campo de visão (FoV) para cobrir os alvos.

nismos para aumentar a cobertura da rede após uma implantação de forma aleatória. Portanto, o surgimento das câmeras PTZ, no contexto das redes de sensores visuais, foi de grande importância para o aumento da cobertura em uma variedade de aplicações. Além disso, algumas pesquisas também se utilizam dos benefícios das câmeras PTZ para aumentar a redundância visual e a disponibilidade das RSVSF [Costa e Guedes 2014].

Um estudo recente também aproveita-se das características das câmeras PTZ para abordar os desafios de aplicações nas quais determinados alvos têm maior prioridade de monitoramento que outros [Costa e Guedes 2014]. Nestas aplicações, o posicionamento do campo de visão (FoV) dos sensores deve priorizar os alvos mais importantes quando não for possível a cobertura de todos os alvos. A Figura 2.13 mostra uma rede de sensores visuais sem fio com alvos que apresentam diferentes níveis de prioridades, sendo $P : 1$ os alvos com maior prioridade. Nesses casos, quando não for possível cobrir todos os alvos simultaneamente, os alvos mais prioritários devem ter preferência de cobertura para não comprometer tanto os resultados da aplicação.

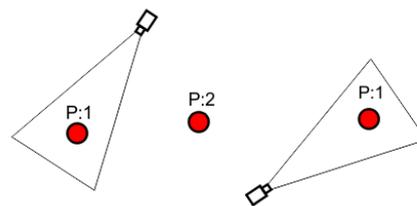


Figura 2.13: Rede de sensores visuais com prioridade de alvos.

2.3.2 Otimização de Cobertura

Várias propostas na literatura buscam desenvolver algoritmos de troca de informação entre os nós visuais da rede de sensores para que seja possível determinar a melhor orientação das câmeras, visando obter uma maior cobertura dos alvos de interesse ou da região monitorada [Yen 2013] [Fusco e Gupta 2009] [Alaei e Barcelo-Ordinas 2009]. Esses algoritmos de otimização de cobertura também tentam minimizar os efeitos das oclusões provocadas por obstáculos espalhados ao redor do ambiente e evitar a sobreposição da região de cobertura entre os sensores visuais para cobrir mais regiões com o menor número de sensores [Almalkawi et al. 2010] [Tezcan e Wang 2008].

Um algoritmo distribuído foi desenvolvido por [Tezcan e Wang 2008] para detectar a cobertura de nós visuais e determinar sua melhor orientação para aumentar a região de cobertura da RSVSF e reduzir as áreas com sobreposição de FoV entre os sensores, conforme ilustra a Figura 2.14. Esse algoritmo considera que cada sensor visual conhece sua localização, através do uso de alguma técnica de localização, e que sua câmera pode alterar seu FoV horizontalmente (câmera *pan*) através de ajustes na sua orientação, em relação ao ambiente implantado. O algoritmo inicia com cada sensor enviando mensagens HELLO em *broadcast* entre os vizinhos para trocar informações sobre a localização e o atual campo de visão (FoV) dos outros sensores. Com essas informações, cada sensor visual identifica a melhor orientação para seu campo de visão realizando uma série de até três testes. Após o ajuste da melhor orientação, o sensor envia uma mensagem para atualizar seus vizinhos com a informação de seu novo FoV.

No primeiro teste, o algoritmo faz com que cada sensor visual realize uma varredura panorâmica de 360° com sua câmera e encontre um FoV sem obstruções e sem regiões de cobertura sobrepostas com a de outros sensores. Caso não seja encontrado esse FoV, o sensor executa o teste *Neighbor-Distance*. Esse segundo teste avalia se um sensor tem um FoV livre de obstruções e com a menor região de sobreposição em relação a seus vizinhos. Por fim, não encontrando um FoV que atenda ao teste *Neighbor-Distance*, o sensor executa o teste *Obstacle-Distance* para encontrar um FoV que evite grandes oclusões provocadas por obstáculos muito próximos do sensor de forma a obter a maior região de cobertura possível mesmo na presença de obstáculos [Tezcan e Wang 2008].

A Figura 2.14 mostra o aumento que há na região de cobertura de uma RSVSF quando esse algoritmo de otimização de cobertura é usado, Figura 2.14(b), em comparação com um ajuste aleatório na orientação dos sensores, Figura 2.14(a). Percebe-se uma redução na sobreposição da área de cobertura das câmeras e uma melhor distribuição do FoV dos sensores em torno do ambiente levando, assim, a uma melhora da região de cobertura em torno de 8,79%. Essa simulação foi realizada levando em consideração a implantação de 50 sensores visuais de forma aleatória e a inclusão de quatro obstáculos na região de monitoramento.

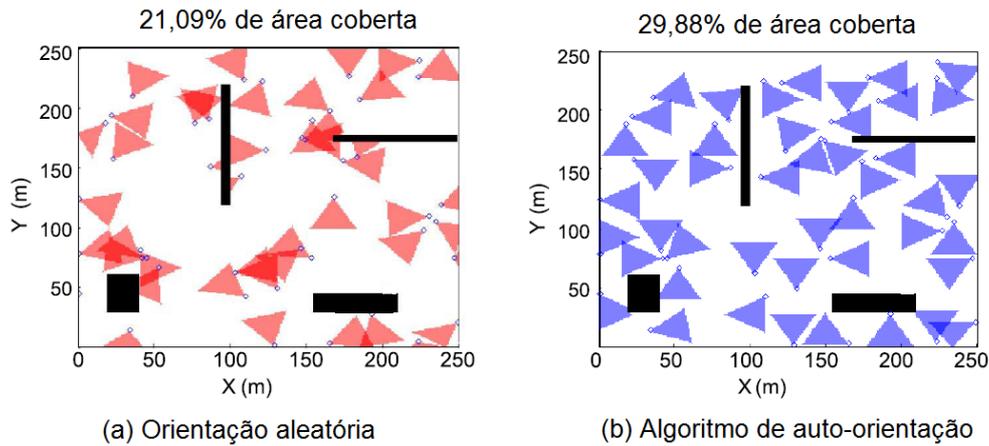


Figura 2.14: Aumento da região de cobertura em uma RSVSF [Tezcan e Wang 2008].

O trabalho de [Yen 2013] propôs um algoritmo heurístico, nomeado de FoVIC, para otimizar a cobertura das redes de sensores visuais. A ideia básica desse algoritmo é implantar um sensor de cada vez em uma orientação que possa abranger o máximo de alvos ainda não cobertos. Essa abordagem considera que os sensores visuais só podem ser implantados em uma quantidade limitada de locais, pré-fixados, em torno do ambiente. Então, os sensores vão sendo implantados, um a um, até cobrir todos os alvos ou não haver mais locais para implantar os sensores. Após a implantação de cada sensor visual, é verificado se há outros sensores implantados que podem ser removidos sem prejudicar a cobertura dos alvos. Ou seja, o algoritmo busca os sensores que apresentam alvos que já são cobertos por outros sensores e que podem ser retirados do ambiente sem impactar na cobertura total dos alvos. Ao remover esses nós redundantes, o algoritmo consegue cobrir todos os alvos com o menor número de sensores visuais reduzindo, assim, o custo final da aplicação. A Figura 2.15 ilustra o funcionamento do algoritmo FoVIC.

A Figura 2.15(a) apresenta uma região com diversos alvos que deverão ser cobertos pelos sensores visuais. Usando o algoritmo FoVIC, cada sensor implantado tem a orientação de seu FoV ajustada para cobrir o máximo de nós ainda não cobertos, como pode ser visto nas Figuras 2.15(b), 2.15(c), 2.15(d), 2.15(e) e 2.15(g). Quando for possível remover algum nó já implantado sem alterar a cobertura dos alvos, o algoritmo FoVIC realiza a remoção, como é mostrado nas Figuras 2.15(f) e 2.15(h). O algoritmo finaliza sua execução quando todos os alvos estão cobertos, Figura 2.15(h). Os resultados obtidos com o algoritmo FoVIC, através de experimentos, mostram que quanto maior o ângulo de visão dos sensores visuais, mais alvos podem ser cobertos e menos sensores são necessários para cobrir todos os alvos de interesse da aplicação [Yen 2013]. Entretanto, esse algoritmo apresenta algumas limitações uma vez que restringe os locais onde os sensores podem ser implantados e também propõe a remoção dos sensores visuais redundantes para redução de custos, algo que nem

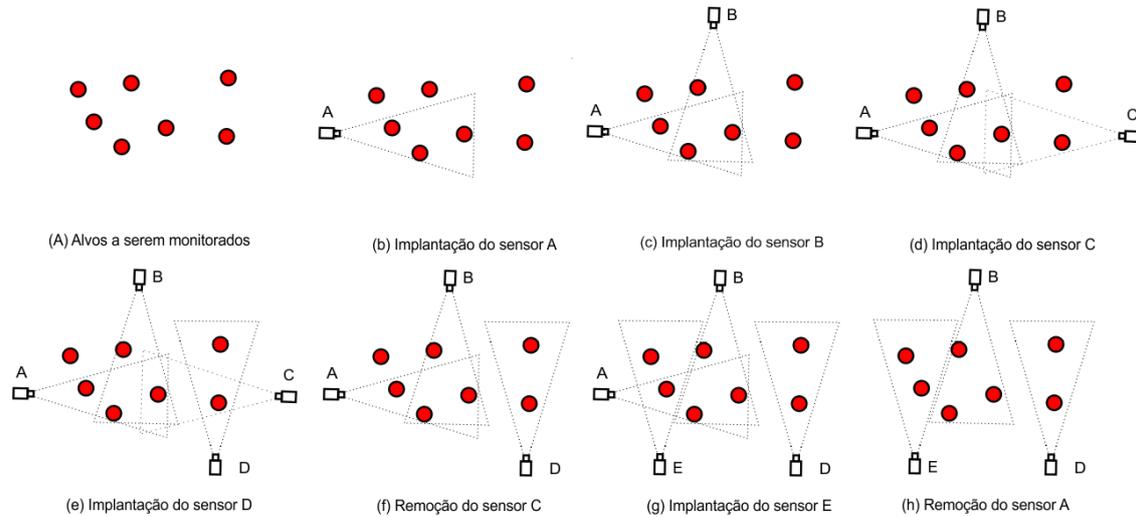


Figura 2.15: Funcionamento do algoritmo FoVIC para otimização da cobertura.

sempre será possível e irá depender da aplicação.

Outro algoritmo que busca maximizar a região de cobertura de sensores com limitado campo de visão foi desenvolvido por [Fusco e Gupta 2009]. A proposta tem o objetivo de resolver o problema de selecionar o menor número de sensores e determinar a melhor orientação para cada um deles de forma a cobrir k vezes um determinado conjunto de alvos ou região, sendo esse problema de grande complexidade e considerado NP-Difícil mesmo quando limitações são impostas. Assim, uma solução aproximada para esse problema foi proposta por [Fusco e Gupta 2009], que desenvolveu um algoritmo guloso, porém com determinadas garantias de desempenho. Tais garantias asseguram que, se houver uma solução na qual M sensores possam cobrir k vezes todos os alvos, então, o algoritmo guloso desenvolvido oferece uma solução para cobrir k vezes pelo menos metade dos alvos usando, no máximo, $M \log(k|C|)$ sensores, onde $|C|$ é o número máximo de alvos cobertos por um sensor.

Esse algoritmo trabalha realizando iterações, em cada iteração é avaliado todos os d sensores que ainda não foram selecionados nas iterações anteriores. Esses d sensores então são testados em todas as orientações possíveis de forma a encontrar qual deles consegue cobrir a maior quantidade de alvos ainda não cobertos k vezes, e em qual orientação de seu FoV isso acontece. O algoritmo termina quando todos os alvos são cobertos k vezes. Ao término de cada iteração é fixada a orientação de um sensor e iterações posteriores não poderão alterar essa configuração. Os sensores não selecionados, após o término do algoritmo, permanecem inativos para preservar a energia [Fusco e Gupta 2009]. A Figura 2.16 apresenta o funcionamento do algoritmo guloso para cobrir todos os alvos duas vezes($k = 2$).

Considerando a distribuição de vinte alvos e de quatro sensores visuais em um ambiente, como ilustrado na Figura 2.16(a), a primeira iteração do algoritmo guloso

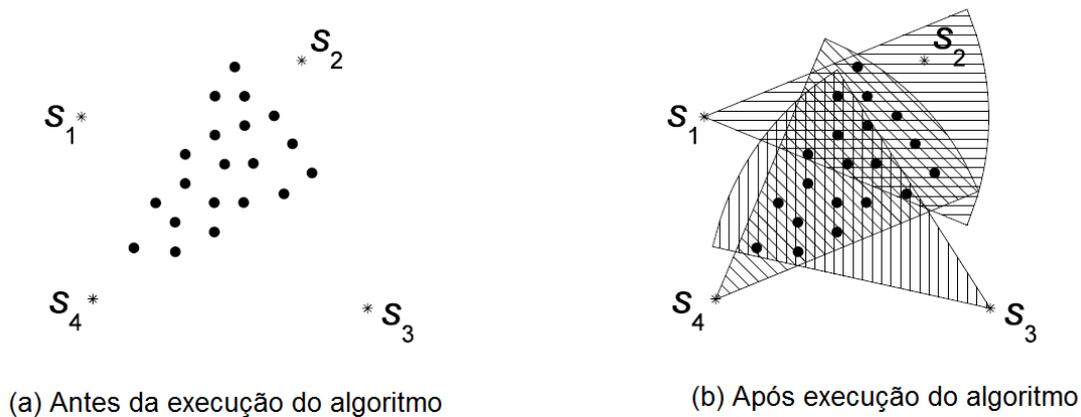


Figura 2.16: Execução do algoritmo guloso para cobrir todos os alvos 2 vezes ($k = 2$) [Fusco e Gupta 2009].

seleciona e ajusta a orientação do sensor S_4 . Isso acontece por que este sensor é o que pode cobrir um maior número de alvos (todos os vinte alvos) entre os sensores que há no ambiente. Em seguida, é selecionado e orientado o sensor S_3 , que consegue cobrir 15 alvos pela segunda vez. Por fim, S_1 é selecionado para cobrir os 5 alvos restantes pela segunda vez. Com isso, o algoritmo atinge o objetivo da aplicação, que era cobrir todos os alvos duas vezes, e o sensor S_2 é desativado para conservar sua energia, como mostra a Figura 2.16(b).

Nas diversas simulações realizadas por [Fusco e Gupta 2009], o algoritmo guloso precisou selecionar apenas 25% mais sensores do que o mínimo teórico para cobrir k vezes todos os alvos. Porém, a depender da disposição dos sensores e dos alvos, o algoritmo guloso pode apresentar problemas e não cobrir todos os alvos, mesmo quando teoricamente isso for possível. Além disso, diferentemente dos demais algoritmos de otimização de cobertura, esse algoritmo guloso oferece a possibilidade de aumentar a redundância visual sobre os alvos monitorados quando definido um $k > 1$. Mas, o algoritmo não tem por objetivo maximizar a cobertura redundante uma vez que, satisfeita a cobertura de todos dos alvos k vezes, os demais sensores visuais são desativados.

Com isso, os algoritmos de otimização da cobertura, apesar de ampliar a região de cobertura da RSVSF, não se preocupam com questões relacionadas à disponibilidade das redes. O fato da otimização da cobertura, na maioria dos casos, buscar reduzir as regiões de sobreposição entre o FoV dos sensores visuais torna as aplicações das RSVSF mais vulneráveis a problemas de falhas dos nós e esgotamento da energia, impactando diretamente no tempo de vida da rede. Isso acontece justamente pela falta de redundância visual provocada pela maioria dos algoritmos de otimização de cobertura, fazendo com que uma determinada área monitorada fique dependente de apenas um sensor visual que pode falhar ou esgotar sua energia e provocar a

indisponibilidade de dados para a aplicação e, com isso, reduzir a qualidade dos resultados obtidos.

2.3.3 Redundância Visual

Alguns trabalhos recentes tentam desenvolver algoritmos de cobertura para ampliar a redundância visual entre os sensores e com isso obter diferentes perspectivas dos alvos ou da região monitorada. Estas diferentes perspectivas possibilitam a aplicação obter mais detalhes do que está sendo monitorado, dando maior precisão aos resultados obtidos, mesmo que para isso seja necessária a implantação de mais sensores. Outra vantagem da sobreposição da região de cobertura pelos sensores visuais é o fato de aumentar a disponibilidade da RSVSF. O aumento da disponibilidade permite que mesmo com a perda de alguns sensores (por mau funcionamento ou esgotamento de energia) a rede continue funcionando atendendo às necessidades da aplicação ao ter outros sensores visuais (os sensores redundantes) que podem continuar monitorando a região ou os alvos de interesse da aplicação devido a sobreposição visual que há [Costa e Guedes 2014] [Alaei e Barcelo-Ordinas 2009].

A redundância nas RSVSF pode ser considerada como a sobreposição do campo de visão de dois ou mais sensores visuais, o que pode indicar que estes sensores estão obtendo informações equivalentes [Costa e Guedes 2010]. Porém, por se tratar de informações visuais, essa equivalência pode não ser adequada para determinadas aplicações. Isso acontece, por exemplo, quando a orientação dos sensores visuais sobrepostos são tão diferentes a ponto de um dos sensores capturar uma imagem do alvo em um determinado ângulo que não contribui para o objetivo da aplicação. Em resumo, o problema de cobertura redundante está concentrado sobre como uma área de interesse é melhor coberta por sensores visuais sem fio e, para áreas densamente implantadas, como o tempo de vida da rede pode ser prolongado pela troca de sensores ligados e desligados, aumentando o tempo de coleta de dados no ambiente monitorado [Costa e Guedes 2014].

Um algoritmo que se aproveita da cobertura redundante para preservar a energia das redes de sensores visuais sem fio foi desenvolvido por [Alaei e Barcelo-Ordinas 2009]. Este algoritmo se baseia na cooperação entre sensores visuais redundantes para detecção e monitoramento de alvos. Nessa proposta, a rede é agrupada com base na sobreposição das áreas de cobertura dos sensores visuais, formando um *cluster*. Para calcular a região com sobreposição de cobertura, o algoritmo proposto determina os vértices do FoV de cada sensor considerando a região de cobertura visual como um triângulo isósceles. A localização dos sensores visuais representa um dos três vértices do triângulo e os demais vértices são encontrados com base no ângulo de visão, no raio de alcance e na orientação da câmera acoplada ao sensor. A área de sobreposição da região de cobertura entre dois sensores visuais deve ser maior do que um certo limiar para que os dois sensores possam pertencer ao mesmo *cluster*. Assim, um *cluster* consiste de um subconjunto de sensores visuais com alta sobreposição

da região de cobertura. A Figura 2.17 apresenta a parte desse algoritmo que é responsável por criar e determinar quais sensores devem fazer parte de um *cluster*.

```

Cluster Membership ( )
1  For each  $S_i$  // un-clustered multimedia node //
2    create an empty cluster ( $C_k$ )
3    mark  $S_i$  as a clustered multimedia node
4     $S_i$  is the first member of cluster  $C_k$ 
5    for each  $S_j$  // un-clustered multimedia node //
6      find intersection polygon of FoVs of  $S_i, S_j$ 
7      compute  $D_{ij}$  //overlapped area between  $S_i, S_j$ //
8      If ( $D_{ij} \geq$  clustering threshold)
9        mark  $S_j$  as a clustered multimedia node
10        $S_j$  is a member of  $C_k$ 
11     endif
12   endfor
13 endfor

```

Figura 2.17: Algoritmo responsável em montar os *clusters* [Alaei e Barcelo-Ordinas 2009].

Esse algoritmo considera que os sensores são implantados de forma aleatória e em grande quantidade, e que cada sensor visual conhece sua localização e orientação, através do uso de alguma técnica de localização. As câmeras acoplados aos sensores são fixas, ou seja, não podem ajustar seu FoV. Tais tipos de sensores são mais baratos do que sensores visuais do tipo PTZ, reduzindo assim o custo da RSVSF. Visando reduzir a complexidade do algoritmo, o FoV dos sensores visuais é aproximado para a área de um triângulo isósceles, de forma similar como acontece com outras técnicas já apresentadas anteriormente. Por fim, para preservar a energia da RSVSF, cada *cluster* programa seus sensores para serem ativados de forma sequencial e intercalada ao longo do tempo e, em seguida capturam, processam e transmitem uma imagem ao *sink* da rede. Logo depois, voltam ao estado inativo e aguardam seu próximo período de atividade conforme a ordem estabelecida pelo *cluster* [Alaei e Barcelo-Ordinas 2009]. Dessa forma, quanto mais sensores tiver um *cluster*, maior será o intervalo de tempo no qual os sensores ficarão inativos conservando energia e, conseqüentemente, aumentando a disponibilidade da rede. Além disso, cada sensor visual dentro de um *cluster* captura uma imagem do alvo monitorado em uma perspectiva diferente, permitindo que a aplicação possa ter diferentes imagens do alvo monitorado para obter resultados mais precisos.

Outro trabalho, desenvolvido por [Costa et al. 2014b], propõe um algoritmo guloso, nomeado de CPGA, com o objetivo de aumentar a redundância das RSVSF para o monitoramento de alvos. Nesse trabalho, é considerado que os alvos são estáticos

e que os sensores visuais implantados podem ajustar seu campo de visão (FoV) para melhorar a cobertura e aumentar a redundância visual. Também é levado em conta que alguns alvos podem ser mais importantes que outros para determinada aplicação, por isso, o algoritmo apresenta uma forma de priorizar a cobertura desses alvos. Segundo [Costa et al. 2014b], esse é o primeiro trabalho a abordar o problema de maximizar a redundância usando o maior número de sensores implantados.

O algoritmo guloso proposto inicia sua execução tentando encontrar, para cada sensor visual, a melhor orientação, dentre as disponíveis, capaz de cobrir a maior quantidade de alvos ainda não cobertos. Para isso, o algoritmo usa as informações sobre a localização e orientação de todos os sensores bem como a posição dos alvos no ambiente e retorna, ao final de sua execução, a melhor orientação para todos os sensores visuais implantados. Essas informações devem então ser transmitidas, através da rede, para os sensores ajustarem seu FoV e cobrir os alvos com a maior redundância possível. O algoritmo permite ainda priorizar a redundância sobre os alvos mais importantes para a aplicação. Dessa forma, os alvos com maior prioridade poderão ter preferência na cobertura pelos sensores visuais, conforme às necessidades de cada aplicação. A Figura 2.18 apresenta a melhor distribuição dos sensores visuais após a execução desse algoritmo guloso.

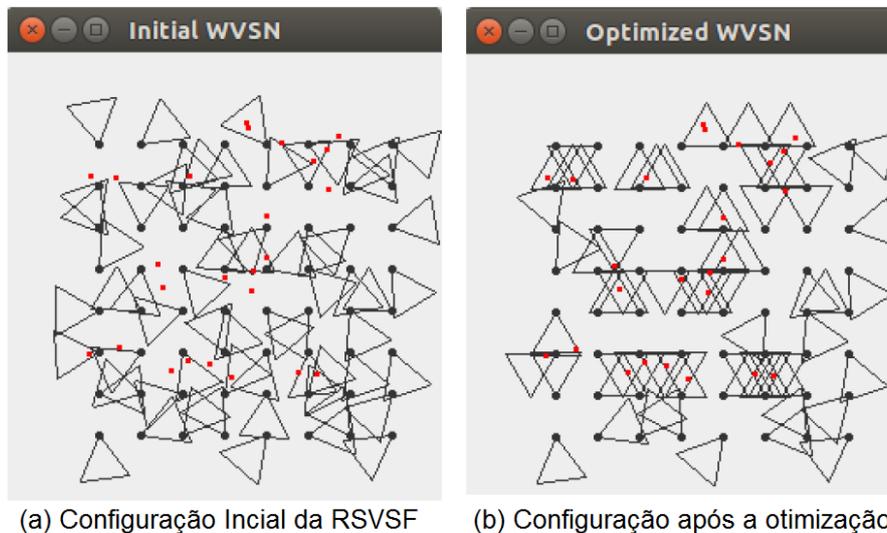


Figura 2.18: Simulação do algoritmo guloso para melhoria da redundância [Costa et al. 2014b].

A simulação realizada consiste de um ambiente com 64 sensores visuais e 25 alvos implantados de forma aleatória. Na configuração inicial da RSVSF existe 12 alvos não cobertos, como mostra a Figura 2.18(a). Após a execução do algoritmo guloso os sensores ajustam seu FoV para otimizar a cobertura e maximizar a redundância, como ilustrado na Figura 2.18(b). Com isso, todos os 25 alvos passam a ser cobertos e, na maioria dos casos, por mais de um sensor visual. Esse aumento na redundância visual dos alvos monitorados permite, por exemplo, que aplicações de localização e de

reconhecimento facial possam obter múltiplas visões do mesmo alvo para melhorar a qualidade dos resultados obtidos [Costa et al. 2014b].

Porém, o desenvolvimento de um algoritmo guloso, visando o aumento da cobertura das redes de sensores visuais sem fio, apresenta uma desvantagem. A depender da disposição inicial dos sensores visuais e dos alvos a serem monitorados, é possível que o algoritmo guloso não consiga cobrir todos os alvos da aplicação, mesmo que isso seja teoricamente viável. A Figura 2.19 ilustra quando isso pode acontecer. Percebe-se que, após a execução do algoritmo guloso, um dos alvos não foi coberto por nenhum sensor visual, como mostra a Figura 2.19(b). Isso acontece devido a características próprias de um algoritmo guloso, que busca sempre a melhor orientação para cada sensor visual de forma independente.

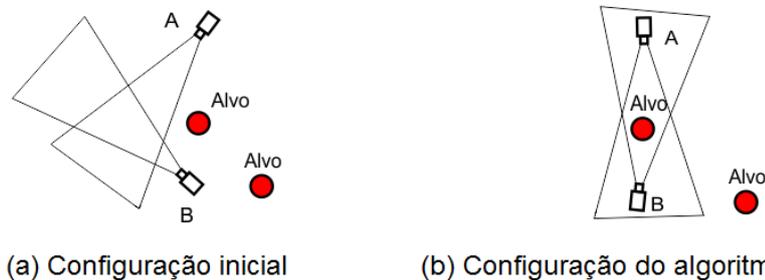


Figura 2.19: Problema do algoritmo guloso na cobertura de alvos.

Por fim, em aplicações críticas, sensores visuais com câmeras PTZ podem ser implantados de forma aleatória sobre um ambiente e em seguida, um algoritmo híbrido pode ser construído para otimizar a cobertura da RSVSF, de forma a cobrir todos os alvos, e também maximizar a cobertura redundante através do ajuste do FoV dos sensores implantados. Isso permitiria aumentar a precisão dos dados coletados, melhorar a disponibilidade da rede, tornar possível o uso das RSVSF em mais aplicações de alta relevância e oferecer rápida resposta a eventos críticos, além de incentivar o surgimento de novas aplicações. Esse é um campo de pesquisa ainda pouco explorado e este trabalho tem o objetivo de contribuir para a ascensão dessa área de pesquisa, propondo o desenvolvimento de algoritmos para atender a essas necessidades.

Capítulo 3

Metodologia

Para o desenvolvimento deste trabalho, a metodologia utilizada foi fundamentada em redes de sensores sem fio, redes de sensores visuais e cobertura visual. Em seguida, foram feitas pesquisas mais aprofundadas com relação a otimização de cobertura, disponibilidade e redundância em redes de sensores visuais sem fio. Além disso, técnicas e algoritmos para garantir a otimização da cobertura e o aumento da redundância visual dessas redes foram analisados com o objetivo de identificar as oportunidades existentes que podem ser exploradas.

O próximo passo foi determinar qual algoritmo de otimização de cobertura seria adaptado de forma a atender os objetivos propostos. Em seguida, as adaptações foram realizadas visando a maximização da redundância das redes de sensores visuais sem fio para o monitoramento de alvos, buscando melhorar não só a disponibilidade dessas redes como também permitir diferentes perspectivas dos alvos monitorados. Após as adaptações, um novo algoritmo ainda foi desenvolvido usando uma abordagem diferenciada das apresentadas na literatura.

Posteriormente, foram realizados testes, análises e comparações dos resultados obtidos para medir a eficácia das técnicas desenvolvidas bem como identificar suas limitações. Para isso, simuladores específicos para os testes foram usados e adaptados de acordo com a necessidade dos resultados que se desejou obter. As comparações têm o objetivo de mostrar os benefícios das abordagens desenvolvidas em termos de aumento da cobertura redundante das RSVSF.

A seguir serão apresentados os detalhes envolvidos no desenvolvimento deste trabalho, incluindo a descrição formal do problema, as técnicas, procedimentos e ferramentas utilizadas para solucionar o problema e a forma como os testes foram realizados e como os resultados foram extraídos dos testes. Assim, o objetivo desse capítulo é deixar claro a proposta do trabalho, como ele foi modelado, tratado e solucionado e quais parâmetros foram avaliados para gerar os resultados obtidos.

3.1 Formalização do Problema

O objetivo principal do trabalho é adaptar e desenvolver novos algoritmos para a maximização da cobertura redundante em redes de sensores visuais sem fio de forma que os sensores visuais possam obter a melhor orientação para cobrir o máximo de alvos da aplicação e com a maior redundância visual possível. Assim, alguns conceitos serão formalmente definidos e o escopo do trabalho será delimitado para que seja possível compreender as soluções criadas bem como os resultados obtidos.

O primeiro conceito a ser definido é o de "Cobertura". Em relação a redes de sensores visuais sem fio, a cobertura trata-se da capacidade de um sensor poder visualizar ou detectar com nitidez, através do FoV de sua câmera, objetos que são considerados alvos para uma aplicação específica. Já o conceito de redundância pode ter um significado diferente de acordo com cada área do conhecimento e diversas podem ser as formas de avaliação da redundância no contexto das redes de sensores visuais sem fio. Portanto, a fim de buscar padronizar esse conceito dentro do escopo desse trabalho, será considerado "Redundância Visual" ou "Cobertura Redundante" quando dois ou mais sensores estiverem monitorando (visualizando) um mesmo alvo, independente da perspectiva visual que cada sensor esteja obtendo em relação ao alvo.

Assim, a cobertura redundante trata-se da capacidade da rede de sensores visuais ter alvos cobertos por mais de um sensor visual, ou seja, ter dois ou mais sensores cobrindo determinado alvo da aplicação. Dessa forma, mesmo com a falha de um sensor, não haverá interrupção no monitoramento do alvo, pois há outro sensor (o sensor redundante) realizando a cobertura. Porém, a depender do cenário e da distribuição dos alvos e sensores visuais nem sempre será possível fazer com que todos os alvos tenham uma cobertura redundante. Então, nesse contexto, o importante é melhorar o ajuste das câmeras dos sensores para, sempre que possível, fornecer a redundância visual sobre o maior número de alvos da rede.

Considere S um conjunto de n sensores de uma rede que é representado como $S = \{s_1, s_2, \dots, s_n\}$, onde n é a quantidade de sensores da aplicação. Agora suponha T um conjunto de m alvos espalhados pelo cenário da aplicação, representado como $T = \{t_1, t_2, \dots, t_m\}$. Pode-se definir também os alvos que um sensor i qualquer está cobrindo em sua atual orientação como um subconjunto do conjunto T da seguinte forma: $C_{s_i} = \{target \mid target \in T \text{ e está dentro do } FoV \text{ do sensor } i\}$, sendo i um inteiro entre 1 e n .

A partir disso, podemos verificar que todos os alvos com cobertura redundante (com dois ou mais sensores cobrindo-os) podem ser definidos pelo conjunto CR como:

$$CR = \{alvo \mid alvo \in (C_{s_1} \cap C_{s_2}) \cup (C_{s_1} \cap C_{s_3}) \cup \dots \cup (C_{s_1} \cap C_{s_n}) \cup (C_{s_2} \cap C_{s_3}) \cup (C_{s_2} \cap C_{s_4}) \cup \dots \cup (C_{s_2} \cap C_{s_n}) \cup \dots \cup (C_{s_{(n-1)}} \cap C_{s_n}) \}$$

Nesse trabalho, o conceito de cobertura redundante aparece muito associado ao termo maximização. A maximização da cobertura redundante, nesse caso, representa a obtenção da maior cobertura redundante possível sobre os alvos da rede,

dados as limitações impostas pelos sensores visuais, os alvos e a distribuição desses elementos no cenário da aplicação. É importante ressaltar também que considera-se que uma maior cobertura redundante é obtida por determinado algoritmo quando este, no geral, consegue ter os sensores visuais da rede cobrindo mais vezes os alvos da aplicação do que outro algoritmo usado para comparação. Ou seja, o algoritmo que, no total, possibilitar que os alvos sejam cobertos uma maior quantidade de vezes pelos sensores será considerado o algoritmo que apresenta maior cobertura redundante.

Dessa forma, é possível calcular formalmente a Máxima Cobertura Redundante (MCR) alcançada por um algoritmo de maximização da seguinte maneira:

$$MCR = |C_{s1}| + |C_{s2}| + |C_{s3}| + \dots + |C_{sn}|$$

Sendo que $|C_{s1}|$ representa a cardinalidade do conjunto de alvos cobertos pelo sensor 1, ou seja, a quantidade de elementos desse conjunto e n é a quantidade de sensores da aplicação. Assim, a soma das cardinalidades de todos os conjuntos que representam a cobertura de cada sensor permite quantizar quantas vezes todos os alvos foram cobertos pelos sensores incluindo as coberturas redundantes.

Em termos de avaliação, a redundância visual obtida por um algoritmo de maximização da cobertura redundante será calculada em função da quantidade de vezes na qual todos os alvos da aplicação foram cobertos. Por exemplo, considerando um cenário de teste com 20 alvos e 50 sensores, um algoritmo "A", de maximização da cobertura redundante, pode cobrir os 20 alvos da aplicação 60 vezes, enquanto que um outro algoritmo "B", no mesmo cenário de teste, pode cobrir os 20 alvos por 75 vezes. Com isso, tanto o algoritmo "A" quanto o algoritmo "B" cobriram todos os alvos da aplicação, porém, o algoritmo "B" apresentou uma cobertura visual redundante maior que o algoritmo "A" uma vez que cobriu mais vezes todos os alvos da aplicação. Essa é a forma de avaliação escolhida para medir a eficácia dos algoritmos desenvolvidos, porém, não é a única forma de avaliação disponível e outras formas de mensurar os resultados obtidos podem ser usadas a partir dos dados gerados com os resultados dos testes. As comparações entre os algoritmos desenvolvidos são feitas calculando o aumento percentual na cobertura redundante que um algoritmo oferece em relação a outro. Assim, usando os dados do exemplo apresentado, o algoritmo "B" seria 25% melhor que o algoritmo "A".

Para o desenvolvimento dos algoritmos de maximização da cobertura redundante, algumas considerações foram feitas para delimitar o escopo do problema. Assim, considera-se que os sensores visuais são integrados com câmeras PTZ, ou seja, podem ajustar sua orientação para mudar o seu campo de visão (FoV) permitindo a otimização da cobertura da rede de sensores. Também foi definido que os alvos da aplicação pudessem ser posicionados em qualquer local do cenário, permitindo a sua implantação de forma aleatória, cabendo aos sensores visuais fazerem o reconhecimento do ambiente e a detecção dos alvos. Além disso, delimitou-se que não cabe a este trabalho o desenvolvimento de técnicas de computação visual para a detecção de

alvos. Assim, foi considerado que, para isso, pode-se utilizar alguma das abordagens já disponíveis na literatura.

Neste trabalho, também não iremos nos preocupar com as perspectivas nas quais um determinado alvo está sendo monitorado, pois nosso interesse inicial é investigar formas de aumentar a redundância visual sem levar em consideração características visuais dos alvos. Portanto, nas simulações, os alvos serão considerados de forma genérica, porém, trabalhos posteriores podem aprimorar a abordagem para solucionar problemas mais específicos, como o reconhecimento facial, caracterização de objetos, entre outras aplicações onde a perspectiva visual, ou seja, a direção na qual o objeto está sendo visualizado é fator crucial para a aplicação.

Foi considerado também que os alvos podem ter diferentes níveis de prioridades que são representados por um valor inteiro atribuído a cada alvo. No desenvolvimento dos algoritmos de cobertura redundante, os alvos com maior prioridade devem ter preferência na cobertura quando os sensores visuais precisam decidir quais alvos cobrir. A atribuição de prioridades aos alvos é uma maneira de flexibilizar o uso desses algoritmos de cobertura para diversas aplicações. Isso permite que alvos mais importantes para uma determinada aplicação sejam levados em consideração no momento de realizar a cobertura dos alvos da rede.

Por último, é essencial deixar claro que apesar do objetivo do trabalho ser maximizar a cobertura redundante sobre os alvos da aplicação, é importante também evitar que haja alvos descobertos. Assim, os algoritmos desenvolvidos e adaptados buscam sempre manter todos os alvos cobertos, ao mesmo tempo em que busca aumentar a cobertura redundante deles. Sendo que, entre os dois objetivos, a preferência maior é fazer com que todos os alvos sejam cobertos. Torna-se claro que, em alguns casos, isso poderá reduzir a cobertura redundante de alguns alvos, porém considera-se que a cobertura de todos os alvos é necessária para um melhor desempenho da aplicação. Dessa forma, um algoritmo de cobertura redundante ideal é aquele que além de maximizar a cobertura redundante da rede, consegue também cobrir todos os alvos da aplicação pelo menos uma vez.

3.2 Técnicas e Procedimentos

As técnicas utilizadas para o desenvolvimento das adaptações e dos novos algoritmos para maximização da cobertura redundante nas redes de sensores visuais sem fio foram escolhidas baseadas em alguns dos principais trabalhos disponíveis na literatura relacionados a cobertura das RSVSF. Com isso, buscou-se selecionar o que há de mais recente na literatura para construir a base necessária para atingir os objetivos propostos nesse trabalho. Isso envolve, basicamente, determinar as técnicas de modelagem matemática a serem usadas para descrever o problema da cobertura visual dos sensores e analisar os algoritmos já desenvolvidos atualmente para otimização

da cobertura afim de escolher o mais apropriado para realizar as adaptações e obter a maximização da cobertura redundante.

Assim como em muitos trabalhos [Ai e Abouzeid 2006] [Fusco e Gupta 2009] [Alaei e Barcelo-Ordinas 2009] [Yen 2013] [Costa et al. 2014b], limitamos o problema a um ambiente 2D. Com isso, a modelagem matemática utilizada leva em consideração algumas simplificações que permitem facilitar os algoritmos desenvolvidos. Sabe-se que a simplificação reduz a precisão dos resultados para um ambiente real, porém, permite que novos conceitos e técnicas sejam desenvolvidos de forma mais rápida e prática bem como possibilita ter uma noção, mesmo que não exata, dos resultados esperados e da eficácia de cada técnica em relação a outras. Posteriormente, os trabalhos desenvolvidos podem ser adaptados para as condições reais do ambiente de operação.

Além de usar uma modelagem 2D para tratar o problema, o campo de visão (FoV) dos sensores também foi aproximado para o formato de um triângulo isósceles, como já detalhado na revisão bibliográfica. As câmeras consideradas são do tipo PTZ, porém, para um ambiente 2D, apenas a função PAN (que permite o movimento horizontal da câmera) é considerada no desenvolvimento dos algoritmos. Os alvos também foram simplificados; é considerado que eles são objetos de dimensão desprezível e sem forma específica. Assim, não preocupa-se em verificar se o alvo foi parcialmente ou totalmente coberto, considerando que não existe esse meio termo para os algoritmos desenvolvidos. É a partir dessas simplificações e considerações que os algoritmos começaram a ser desenvolvidos.

O primeiro trabalho disponível na literatura que propôs um algoritmo especificamente voltado para aumentar a cobertura redundante sobre alvos em redes de sensores visuais sem fio foi o desenvolvido por [Costa et al. 2014b], cujo funcionamento foi apresentado na revisão bibliográfica. Em resumo, aquele trabalho consistiu no desenvolvimento do algoritmo nomeado de CPGA (Centralized Prioritized Greedy Algorithm) que utiliza uma abordagem de algoritmo guloso para encontrar a melhor orientação de cada sensor visual da rede visando cobrir os alvos com a maior redundância possível. Por se tratar de um algoritmo guloso, o CPGA está susceptível a falhas podendo provocar redução na cobertura dos alvos, o que é considerado um efeito indesejado e compromete o desempenho da rede de sensores visuais sem fio.

Dessa forma, o primeiro passo do nosso trabalho foi o desenvolvimento de melhorias para o algoritmo CPGA buscando solucionar os problemas gerados pelo algoritmo guloso e também criar mecanismos para aumentar ainda mais a maximização da cobertura redundante sobre os alvos da aplicação. Duas melhorias foram propostas para o CPGA, com isso, um algoritmo adaptado foi construído e nomeado de ECPGA (Enhanced Centralized Prioritized Greedy Algorithm). O CPGA já leva em consideração a prioridade dos alvos para a cobertura da rede, assim, o algoritmo adaptado ECPGA e todos os demais desenvolvidos neste trabalho também incorporaram essa funcionalidade do algoritmo CPGA.

Ambos os algoritmos, o original e o adaptado, são algoritmos centralizados, ou seja,

a determinação da melhor orientação para todos os sensores visuais da rede são definidas por um nó central que deve transmitir essas informações para os demais nós. Assim, outra adaptação construída foi a reimplementação do ECPGA como um algoritmo distribuído, nomeado de EDPGA (Enhanced Distributed Prioritized Greedy Algorithm). Um algoritmo distribuído é outra forma de desenvolver algoritmos para as redes de sensores. O diferencial dessa técnica é que todos os sensores visuais trocam informações para que cada um processe os dados e decida a melhor orientação para ajustar sua câmera, criando uma rede que não depende de um nó central.

Além disso, foi implementado um novo algoritmo, utilizando uma abordagem diferente da proposta pelo CPGA, porém, que utiliza também alguns dos conceitos desenvolvidos no CPGA e das melhorias do ECPGA, sendo nomeado de RCMA (Redundant Coverage Maximization Algorithm). Um diferencial desse algoritmo é que ele não utiliza a abordagem gulosa, ou seja, cada sensor visual não necessariamente vai ajustar sua orientação para cobrir o máximo de alvos possíveis. Os detalhes do desenvolvimento das adaptações baseadas no CPGA e do novo algoritmo construído, o RCMA, serão apresentados no capítulo seguinte.

3.3 Ferramentas

Todos os algoritmos e cenários de testes foram implementados utilizando a linguagem de programação Java. Através dessa linguagem de programação foi possível implementar todos os recursos necessários para simular, testar e comparar os diversos algoritmos desenvolvidos. A escolha em utilizar o Java para a realização deste trabalho deve-se, principalmente, ao código fonte do algoritmo CPGA, desenvolvido por [Costa et al. 2014b], está disponível para uso e/ou modificação nessa linguagem de programação. Com isso, não houve a necessidade de reimplementar o algoritmo CPGA, agilizando o processo de desenvolvimento das adaptações propostas como um todo e permitindo maior precisão nas comparações realizadas, por está utilizando a implementação que de fato foi desenvolvida pelo autor.

Assim, foram realizadas as adaptações propostas utilizando a própria estrutura de desenvolvimento do algoritmo CPGA original. Dentre os recursos disponíveis no código fonte do CPGA está a implementação da modelagem matemática, em relação ao ambiente 2D e ao campo de visão (FoV) dos sensores visuais, o algoritmo de verificação de quais alvos podem ser cobertos por determinado sensor e a implementação para simular a distribuição aleatória dos alvos em torno dos cenários de testes. O código fonte do CPGA foi então estudado e modificado para o desenvolvimento das adaptações propostas. Além disso, foi utilizada a mesma estrutura disponível para o desenvolvimento do novo algoritmo, o RCMA, o que permitiu focar o trabalho especificamente no desenvolvimento dos algoritmos de maximização da cobertura redundante.

Junto com o código fonte do algoritmo CPGA, o autor também disponibilizou o código fonte do ambiente de teste, que oferece uma interface visual permitindo não só obter dados dos resultados como também visualizar o funcionamento do algoritmo desenvolvido. Entretanto, apesar das ferramentas disponibilizadas pelo autor serem uma grande contribuição, ainda foi necessário adaptar o código fonte para realizar os testes e comparações de interesse para esse trabalho. As alterações no código foram basicamente relacionadas à obtenção de mais dados dos resultados para realizar a avaliação de desempenho dos algoritmos desenvolvidos e a criação de mais cenários de testes, conforme será apresentado na próxima seção.

3.4 Cenários de Testes

Após o desenvolvimento de cada algoritmo, testes e comparações foram realizados para permitir análises de desempenho em relação ao nível de cobertura redundante alcançado. Além disso, foi verificada a quantidade de alvos cobertos em cada teste, uma vez que cobrir todos os alvos da aplicação também foi considerado um indicador de desempenho. Os testes são feitos especificando a quantidade de sensores e alvos bem como as dimensões do cenário e as características dos sensores visuais, como ângulo de visão e raio de alcance da cobertura visual. Os dados coletados nos testes consistem, basicamente, na quantidade total de alvos cobertos (cobertura total) após o uso do algoritmo e quantas vezes cada um dos alvos foi coberto pelos sensores visuais do cenário. Com isso, será possível realizar análises e comparações de desempenho entre os diferentes algoritmos.

Em todos os testes, os alvos são distribuídos em torno do cenário usando o método *random()* do Java para simular uma implantação aleatória. O método *random()* tem a função de gerar números aleatórios que são usados para posicionar os alvos dentro do cenário de teste. O mesmo é feito para determinar a orientação (direção da câmera) inicial dos sensores visuais, porém, o posicionamento dos sensores visuais no cenário é realizado de uma forma diferente, com o objetivo de testar melhor o desempenho de cada algoritmo. Assim, os sensores são distribuídos uniformemente pelo cenário e igualmente espaçados entre os demais sensores, permitindo avaliar melhor o funcionamento dos algoritmos ao evitar que os sensores fiquem localizados em posições que prejudiquem explorar ao máximo as características de cada algoritmo, consequentemente, afetando a qualidade dos resultados obtidos. Ou seja, caso os sensores sejam posicionados muito próximos um dos outros ou muito afastados dos alvos, a ponto de não ser possível cobri-los, isso irá prejudicar uma efetiva e real avaliação de desempenho dos algoritmos desenvolvidos.

Os cenários de testes foram especificamente criados para avaliar os algoritmos em relação aos indicadores de desempenho já definidos, sendo estes a quantidade total de alvos cobertos e a quantidade de vezes em que os alvos foram cobertos por cada um dos algoritmos utilizados. Para isso, geralmente, os cenários de testes criados

contém mais sensores visuais do que alvos justamente para que seja possível avaliar a redundância visual sobre os alvos da rede quando são utilizados algoritmos de maximização da cobertura redundante. Não seria teoricamente interessante avaliar o desempenho de algoritmos de redundância quando há poucos sensores visuais em relação a quantidade de alvos disponíveis, já que isso faria com que muitos alvos nem fossem cobertos, dificultando uma avaliação mais eficaz dos algoritmos.

Um detalhe importante é que, nos testes realizados, a prioridade atribuída a todos os alvos foi a mesma de forma que não existe alvos mais importantes no qual os algoritmos desenvolvidos tenham que priorizar para realizar a cobertura. Ou seja, todos os alvos, em todos os cenários de testes e para todos os algoritmos testados, foram configurados para terem o mesmo nível de prioridade apesar dos algoritmos desenvolvidos também tratar alvos com diferentes prioridades. A implementação da prioridade nos algoritmos tem o objetivo de tornar o seu uso mais flexível para diversas aplicações, entretanto, apesar da prioridade ter sido tratada nos algoritmos desse trabalho, não foram realizados testes para avaliar essa característica dos algoritmos. Em trabalhos futuros pretende-se dar maior atenção a essa característica de forma a gerar resultados e comparações especificamente relacionadas a alvos com diferentes prioridades.

Para realizar comparações entre os algoritmos desenvolvidos padronizou-se um cenário de teste fixando a quantidade de elementos da rede em 100 sensores visuais e 80 alvos em um cenário de 240m x 240m. Foi padronizado também as características dos sensores visuais, sendo o ângulo de visão (AoV) das câmeras configurado em 60° e o raio de alcance (DoF) em 30 metros. Assim, a cada algoritmo desenvolvido, uma nova comparação foi realizada permitindo acompanhar também o histórico de desempenho dos primeiros algoritmos desenvolvidos e garantir a consistência dos resultados obtidos. Além disso, as comparações são realizadas fazendo 10 testes sobre esse mesmo cenário padronizado, porém, alterando aleatoriamente a posição dos alvos dentro do cenário e a orientação inicial dos sensores visuais. O conjunto de 10 testes aumentou a precisão dos resultados e permitiu calcular uma média do desempenho de cada algoritmo, fornecendo comparações mais confiáveis.

Por fim, após a construção de todos os algoritmos e adaptações, mais três testes foram realizados para completar a avaliação de desempenho das abordagens desenvolvidas. O primeiro teste buscou analisar o comportamento dos algoritmos implementados para diferentes raios de alcance (DoF) dos sensores visuais. O segundo teste consistiu em variar o ângulo de visão (AoV) dos sensores visuais para verificar a influência desse parâmetro no desempenho dos algoritmos. O último teste realizado mostra os resultados obtidos para cada algoritmo quando os sensores visuais são posicionados no cenário de teste usando o método *random()* do Java, simulando uma implantação aleatória dos sensores. Assim, no próximo capítulo deste trabalho, os resultados obtidos com os cenários de testes serão apresentados juntamente com a descrição dos algoritmos desenvolvidos.

Capítulo 4

Algoritmos Propostos

Foram desenvolvidos três algoritmos para a maximização da cobertura redundante. O primeiro algoritmo, nomeado de ECPGA, consiste em uma adaptação do algoritmo CPGA desenvolvido por [Costa et al. 2014b] e busca criar melhorias importantes para o aumento da cobertura e da redundância visual da rede de sensores. O próximo algoritmo desenvolvido, nomeado de RCMA, utilizou abordagens diferentes para tratar o problema da maximização da cobertura, o que possibilitou um significativo aumento de performance na cobertura visual redundante. Por último, foi desenvolvida uma versão distribuída do algoritmo ECPGA, nomeado de EDPGA, para avaliar o desempenho desse tipo de algoritmo em relação aos demais desenvolvidos.

Será apresentada a seguir uma descrição detalhada da construção desses algoritmos e sua forma de funcionamento passo a passo. Além disso, será dado um enfoque nas principais características de cada algoritmo que contribuíram para a obtenção da maximização da cobertura redundante. Posteriormente, os resultados dos testes realizados serão apresentados de forma a verificar o desempenho alcançado em cada algoritmo implementado. Comparações também foram feitas entre os algoritmos desenvolvidos para avaliar as vantagens e desvantagens de cada uma das abordagens utilizadas.

4.1 ECPGA: Um algoritmo Guloso Aprimorado

A primeira adaptação realizada tomou como base o algoritmo guloso desenvolvido por [Costa et al. 2014b], nomeado de CPGA (Centralized Prioritized Greedy Algorithm). Buscou-se, primeiramente, aumentar a quantidade de possíveis regiões ou setores com que esse algoritmo determina a melhor orientação dos sensores visuais. Dessa forma, foi possível ajustar os sensores em uma orientação, não disponível anteriormente, que possa cobrir mais alvos e conseqüentemente aumentar a cobertura redundante da rede. A segunda adaptação feita no algoritmo de [Costa et al. 2014b]

consistiu em tratar os possíveis problemas gerados pelos algoritmos gulosos. Assim, o algoritmo foi adaptado para localizar e cobrir, quando possível, os alvos que não foram cobertos após a execução do algoritmo original. Com a implementação dessas duas melhorias tem-se o algoritmo adaptado nomeado de ECPGA (Enhanced Centralized Prioritized Greedy Algorithm).

4.1.1 1ª Melhoria: Aumento do número de Regiões de Cobertura

O trabalho desenvolvido em [Costa et al. 2014b] considera que os sensores são fixos, ou seja não podem se deslocar após a implantação, porém são equipados com câmeras ajustáveis do tipo PTZ que podem alterar a orientação do seu campo de visão (FoV). Além disso, o algoritmo propõe que haja um conjunto de finitos setores, diferentes entre si, para o ajuste da direção do FoV dos sensores visuais. A quantidade de setores é definida pelo ângulo de visão dos sensores. Para um sensor visual com ângulo de visão de 45° haverá $360^\circ/45^\circ = 8$ diferentes setores no qual o sensor pode ser ajustado, como ilustra a Figura 4.1.

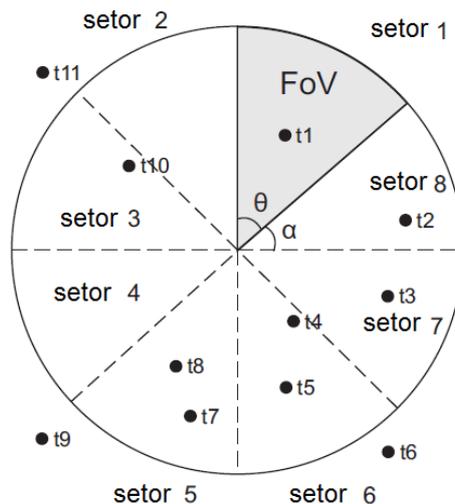


Figura 4.1: Diferentes setores para ajuste da orientação (α) do FoV dos sensores visuais [Costa et al. 2014b].

Entretanto, é importante ressaltar que os sensores visuais do tipo PTZ podem, teoricamente, ajustar a direção do seu campo de visão em infinitas posições diferentes. Porém, computacionalmente se torna custoso tratar as infinitas possibilidades de ajuste das câmeras sendo necessário impor tais limites no desenvolvimento do algoritmo. Outro detalhe importante do algoritmo desenvolvido por [Costa et al. 2014b] é que o campo de visão (FoV) de cada setor cobre apenas uma região totalmente exclusiva dos outros setores, ou seja, não há intersecções de FoV entre os diferentes setores que cada sensor visual pode ser ajustado.

Com base nisso, a primeira adaptação desenvolvida buscou alterar a maneira como os setores são criados de forma que haja uma intersecção na região de cobertura entre os diferentes setores. Isso permite melhorar o ajuste da orientação dos sensores visuais para cobrir mais alvos e conseqüentemente aumentar a redundância visual da rede, que é o principal objetivo desse trabalho. A Figura 4.2 apresenta os novos setores que são criados com a adaptação proposta quando os sensores visuais têm ângulos de visão de 45° .

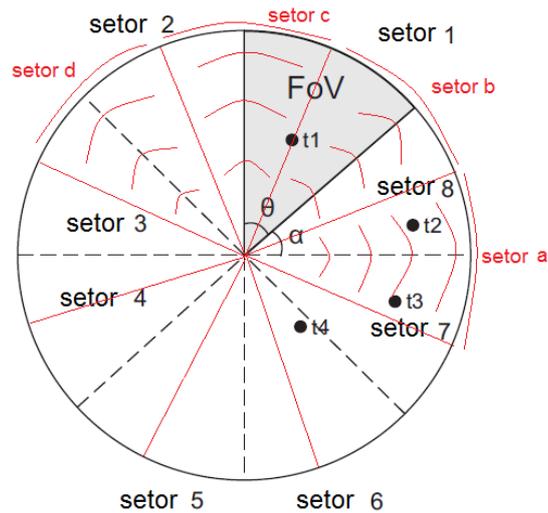


Figura 4.2: Novos setores para ajuste da orientação (α) do FoV dos sensores visuais.

Com base na Figura 4.2 e considerando t_1 , t_2 , t_3 e t_4 como sendo os alvos de interesse da aplicação, é possível perceber que os setores 1, 6, 7 e 8 podem, cada um, cobrir apenas um alvo. Já o setor **a**, criado a partir da adaptação proposta, consegue cobrir dois alvos (t_2 e t_3), sendo um deles coberto pelo setor 7 e outro pelo setor 8. Isso mostra a vantagem em ter novos setores sobrepostos com os demais possibilitando o melhor ajuste das câmeras para cobrir mais alvos. Essa adaptação foi realizada alterando o algoritmo para ajustar o ângulo de orientação (α) dos sensores visuais a cada $\theta/2$ graus, sendo θ o ângulo de visão da câmera do sensor visual. Com isso, os novos setores são sempre formados pela sobreposição de metade da região de cobertura de dois setores do algoritmo original que sejam vizinhos entre si.

O Algoritmo 1 apresenta um pseudocódigo da implementação dessa primeira melhoria para a construção do algoritmo adaptado, o ECPGA. O método *checkViewing*($v, Targets$) é o responsável por encontrar a melhor orientação para um sensor (v) dada uma lista de alvos ($Targets$) implantados no cenário da aplicação. Sendo que a melhor orientação para este método é a que permite um sensor visual cobrir a maior quantidade de alvos ainda não coberto por outros sensores.

O Método *TVT*(*orientation, Target*), também implementado no algoritmo CPGA, é responsável em verificar se determinado alvo (da lista de alvos) pode ser coberto por um sensor (v) em determinada orientação. Já o método *isNotViewed*() é usado para

verificar se determinado alvo ainda não está sendo coberto por nenhum sensor visual. Esse método é importante para fazer com que o Algoritmo 1 consiga encontrar a orientação que cobre a maior quantidade de alvos ainda não cobertos na rede, para cada sensor analisado. É importante ressaltar que essa primeira melhoria também foi usada nos outros algoritmos desenvolvidos nesse trabalho, o RCMA e o EDPGA, devido seu grande potencial de melhoria na determinação da orientação dos sensores visuais.

Algorithm 1 Pseudocódigo da primeira melhoria realizada no algoritmo CPGA

```

1: procedure checkViewing(v, Targets)
2:    $\theta \leftarrow v.getAngle()$ 
3:    $TotalOrientation \leftarrow 2\pi/(\theta/2)$ 
4:    $totalNotViewed \leftarrow 0$ 
5:    $totalPriorities \leftarrow 0$ 
6:    $orientation \leftarrow 0$ 
7:    $targetsViewed[]$  //(lista auxiliar)
8:   for  $j \leftarrow 1$  to  $TotalOrientation$  do
9:      $priorities \leftarrow 0$ 
10:     $notviewed \leftarrow 0$ 
11:     $consideredTargets[]$ 
12:    for  $t \leftarrow 1$  to  $Targets.size()$  do
13:       $target \leftarrow Targets.get(t)$ 
14:      if  $v.TVT((j * \theta/2) \bmod(2\pi), target)$  then
15:        if  $target.isNotViewed()$  then
16:           $notviewed \leftarrow notviewed + 1$ 
17:        end if
18:         $priorities \leftarrow priorities + target.getPt()$ 
19:         $consideredTargets.add(target)$ 
20:      end if
21:    end for
22:    if  $notviewed > totalNotViewed$  then
23:       $totalNotViewed \leftarrow notviewed$ 
24:       $orientation \leftarrow (j * \theta/2) \bmod(2\pi)$ 
25:       $totalPriorities \leftarrow priorities$ 
26:       $targetsViewed \leftarrow consideredTargets$ 
27:    else
28:      if  $notviewed == totalNotViewed$  then
29:        if  $priorities > totalPriorities$  then
30:           $totalPriorities \leftarrow priorities$ 
31:           $orientation \leftarrow (j * \theta/2) \bmod(2\pi)$ 
32:           $targetsViewed \leftarrow consideredTargets$ 
33:        end if
34:      end if
35:    end if
36:  end for
37:  for  $t \leftarrow 1$  to  $targetsViewed.size()$  do
38:     $target \leftarrow targetsViewed.get(t)$ 
39:     $target.setViewed()$ 
40:  end for
41:  return  $orientation$ 
42: end procedure

```

O método $TVT(orientation, Target)$ utilizado no Algoritmo 1 é uma abordagem desenvolvida por [Costa et al. 2014b] para determinar, em ambiente de simulação, se um alvo particular está sendo coberto visualmente por um sensor. A técnica desenvolvida por [Costa et al. 2014b] foi nomeada de *Target Viewing Test (TVT)* e consiste no uso de princípios de trigonometria para verificar se um sensor é capaz de cobrir um alvo na orientação testada. Para isso, esse método precisa conhecer

previamente a posição do sensor e dos alvos testados além de considerar que o FoV dos sensores visuais é aproximado para o formato de um triângulo. Em um ambiente real, essa abordagem deve ser substituída por um algoritmo de computação visual capaz de realizar a detecção dos alvos.

No Algoritmo 1, assim como é realizado no CPGA, a prioridade é avaliada quando, em um sensor visual, há mais de uma orientação que cobre a mesma quantidade de alvos ainda não cobertos por outros sensores. Nesse caso a orientação escolhida para ajustar o sensor visual será aquela que apresentar maior prioridade de cobertura. O cálculo dessa prioridade é feito somando as prioridades individuais de cada alvo que pode ser coberto pelo sensor na orientação avaliada. Com isso, se todas as possíveis orientações cobrir quantidades diferentes de alvos a prioridade não será levada em consideração e a orientação selecionada será a que cobrir a maior quantidade de alvos ainda não cobertos.

4.1.2 2ª Melhoria: Ampliando a Quantidade de Alvos Cobertos

O algoritmo proposto por [Costa et al. 2014b] é um algoritmo guloso que busca a melhor solução para cada sensor da rede. Por isso, em alguns casos, o algoritmo não consegue cobrir todos os alvos da aplicação mesmo quando isso é possível, como podemos ver na Figura 4.3(b). Essa é uma característica típica dos algoritmos gulosos e que pode ser tratada realizando alguma adaptação. Nesse caso, para resolver o problema buscou-se desenvolver um algoritmo que verifique quando um alvo não foi coberto e que realize as medidas necessárias para reverter a situação.

Então, a segunda adaptação consiste em verificar, após a execução do algoritmo original, se há algum alvo ainda não coberto e analisar as possibilidades de cobri-lo. Quando é identificado um alvo ainda não coberto um novo passo do algoritmo desenvolvido nessa adaptação verifica se existe algum sensor visual que pode alterar sua orientação para cobrir esse alvo sem, no entanto, fazer com que alvos anteriormente cobertos percam sua cobertura devido as mudanças de orientação do sensor. Ou seja, um sensor visual só vai mudar sua orientação para cobrir um alvo ainda não coberto caso existam outros sensores com campo de visão (FoV) redundantes que possam manter a cobertura dos alvos já cobertos. A Figura 4.3 ilustra o resultado esperado com a adaptação desenvolvida.

É importante salientar que existem casos em que o algoritmo e as adaptações desenvolvidas não conseguem cobrir determinados alvos. Isso vai depender da quantidade e da disposição dos sensores e alvos ao longo da região de monitoramento. Porém, naqueles casos em que há a possibilidade dos alvos serem cobertos, a adaptação desenvolvida fará com que os sensores visuais necessários alterem sua orientação para cobri-los.

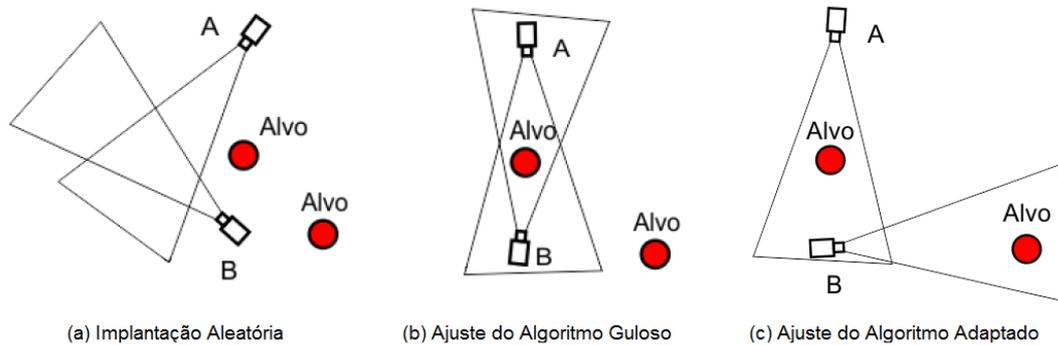


Figura 4.3: Resultado da execução do algoritmo adaptado.

O Algoritmo 2 apresenta o pseudocódigo da implementação dessa segunda melhoria para a construção do algoritmo ECPGA. No início do método *betterCover()* é verificado se há redundância na cobertura dos alvos que determinado sensor está cobrindo (ver linha 6 do Algoritmo 2) para avaliar se ele pode mudar sua orientação para cobrir outros alvos ou não. Para isso, o método *getNumberViewed()* retorna a quantidade de vezes que um determinado alvo está sendo coberto, ou seja, quantos sensores estão cobrindo esse alvo atualmente. Esse método é usado, na linha 6 do algoritmo, para analisar se existem alvos que estão cobertos por apenas um sensor. Caso existam alvos nessas condições, o único sensor que está cobrindo esses alvos não pode alterar sua orientação para cobrir alvos que ainda estão descobertos. Assim, o algoritmo interrompe a análise desse sensor através do comando *return* na linha 7.

Em seguida, o Algoritmo 2 verifica se há alvos não cobertos e tenta mudar a orientação do sensor para fornecer a cobertura visual para esse(s) alvo(s). *TargetCovered*, no pseudocódigo, representa uma lista com os alvos que o sensor visual está atualmente cobrindo. Já o *TargetNotCovered* é uma lista com os alvos que o sensor visual não está cobrindo, porém que está em seu alcance de cobertura visual podendo-os cobrir com alteração de sua orientação. Já o método *getSensorOrientation()* é responsável em informar a orientação no qual o sensor analisado poderá cobrir determinado alvo. Dessa forma, quando um sensor puder alterar sua orientação para cobrir um alvo ainda não coberto, esse método será usado para obter a orientação na qual o sensor terá que se ajustar para passar a cobrir esse alvo.

Algorithm 2 Pseudocódigo da segunda melhoria realizada no algoritmo CPGA

```

1: procedure betterCover()
2:   targetCovered[] ← this.getTargetCoverd()
3:   TargetNotCovered[] ← this.getTargetNotCoverd()
4:   for i ← 1 to TargetCovered.size() do
5:     target ← TargetCovered.get(i)
6:     if target.getNumberViewed() == 1 then
7:       return
8:     end if
9:   end for
10:  if TargetNotCovered.size() == 1 then
11:    target ← TargetNotCovered.get(1)
12:    newOrientation ← target.getSensorOrientation()
13:    this.setOrientation(newOrientation)
14:  else
15:    if TargetNotCovered.size() > 1 then
16:      totalTargetsSameOrient ← 0
17:      bestOrientat
18:      while !TargetNotCovered.isEmpty() do
19:        nTargetsSameOrient ← 0
20:        target ← TargetNotCovered.get(1)
21:        orientat ← target.getSensorOrientation()
22:        listAux[]
23:        for a ← 1 to TargetNotCovered.size() do
24:          targetDiscovered ← TargetNotCovered.get(a)
25:          orientationTD ← targetDiscovered.getSensorOrientation()
26:          if orientationTD == orientat then
27:            nTargetsSameOrient ← nTargetsSameOrient + 1
28:            listAux.add(targetDiscovered)
29:          end if
30:        end for
31:        TargetNotCovered.removeAll(listAux)
32:        if nTargetsSameOrient > contTotal then
33:          bestOrientat ← orientat
34:          totalTargetsSameOrient ← nTargetsSameOrient
35:        end if
36:      end while
37:      this.setOrientation(bestOrientat)
38:    end if
39:  end if

```

O método $ECPGA(DS, Targets)$, apresentado no Algoritmo 3, é o responsável em unir as duas melhorias desenvolvidas para a construção do algoritmo adaptado, o ECPGA, onde os parâmetros DS é a lista de sensores implantados e $Targets$ a lista de alvos da aplicação. Os resultados obtidos através das adaptações desenvolvidas são apresentados a seguir, onde testes e comparações são feitos para constatar as melhorias obtidas com relação a cobertura redundante dos alvos.

Algorithm 3 Pseudocódigo do algoritmo ECPGA

```

1: procedure ECPGA(DS, Targets)
2:   CS[] //(lista de sensores configurados)
3:   for  $i \leftarrow 1$  to DS.size() do
4:     visualSensor  $\leftarrow$  DS.get( $i$ )
5:      $\alpha \leftarrow$  checkViewing(visualSensor, Targets)
6:     visualSensor.setOrientation( $\alpha$ )
7:     CS.add(visualSensor)
8:   end for
9:   for  $i \leftarrow 1$  to CS.size() do
10:    visualSensor  $\leftarrow$  CS.get( $i$ )
11:    visualSensor.betterCover()
12:  end for
13:  return CS
14: end procedure

```

4.1.3 Testes e Comparações

A fim de validar o algoritmo proposto, testes computacionais foram realizados baseados em simulação. O Primeiro teste realizado consistiu em uma simples verificação do desempenho do algoritmo ECPGA, quando comparado com o algoritmo desenvolvido por [Costa et al. 2014b], nomeado de CPGA. Foram testados ambos os algoritmos, partindo inicialmente os sensores visuais de uma orientação aleatória e considerando a mesma disposição, também aleatória, dos alvos para os dois algoritmos. A Figura 4.4 mostra o resultado obtido.

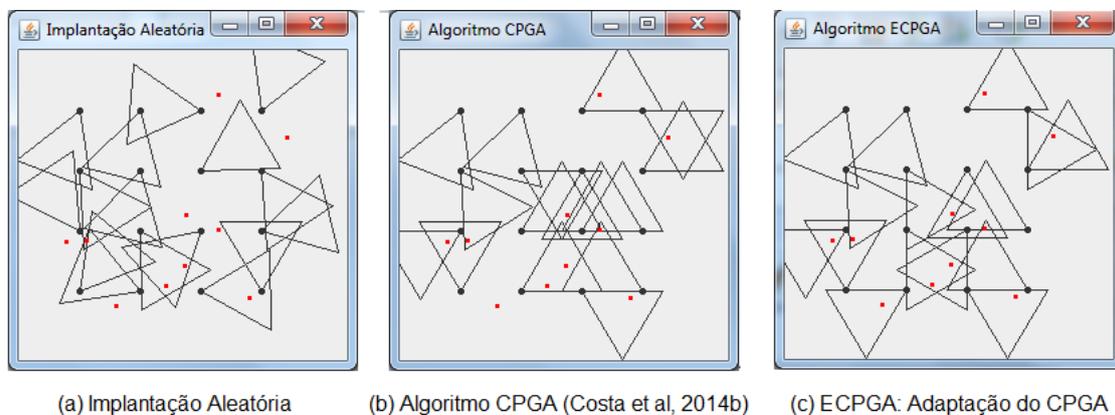


Figura 4.4: Comparação entre os algoritmos CPGA e ECPGA.

Os pontos pretos da Figura 4.4 representam os sensores visuais e os triângulos pretos demarcam a região de cobertura dos sensores visuais, ou seja o FoV de cada sensor. Os algoritmos realizam alterações na orientação dos sensores visuais que por consequência mudam seu FoV para cobrir os alvos. Já os pontos vermelhos representam os alvos de interesse da aplicação que devem ser cobertos pelos sensores visuais. Esse teste foi composto por 16 sensores visuais e 10 alvos. Os sensores visuais tinham

ângulo de visão de 60° , raio de alcance de 30 metros e a região de monitoramento considerada foi de 120m x 120m.

Percebe-se que o algoritmo adaptado (ECPGA) conseguiu cobrir todos os alvos da aplicação enquanto que o algoritmo guloso CPGA deixou de cobrir um alvo. Nesse caso, verifica-se a importância da criação dos setores adicionais juntamente com a verificação final da cobertura, realizado pela segunda melhoria, uma vez que esses foram os fatores determinantes para o aumento da quantidade de alvos cobertos nesse teste.

O próximo teste verificou a capacidade do algoritmo adaptado em aumentar a redundância de alvos cobertos. Para uma melhor visualização do resultado, esse teste foi realizado em um cenário reduzido e envolveu poucos elementos (16 sensores e 10 alvos). O teste considerou o ângulo de visão dos sensores visuais como sendo de 60° , raio de alcance de 30 metros e a região de monitoramento de 120m x 120m. A Figura 4.5 apresenta o resultado obtido no teste.

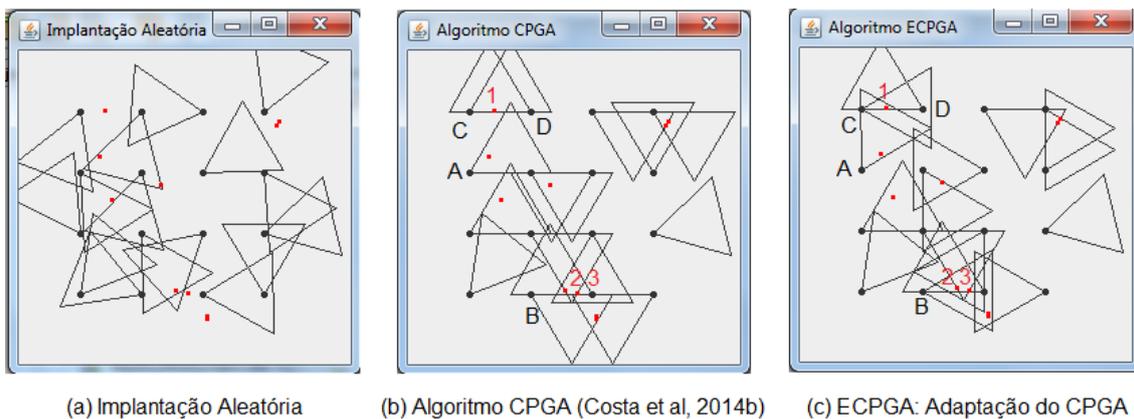


Figura 4.5: Comparação da redundância visual de alvos entre os algoritmos CPGA e ECPGA.

Percebe-se pela Figura 4.5 que tanto o algoritmo CPGA quanto o adaptado fazem a cobertura de todos os alvos. Porém, o algoritmo adaptado, ECPGA, consegue aumentar a redundância visual sobre alguns alvos. No algoritmo original, o CPGA, o alvo de número 1 é coberto por dois sensores (sensor C e sensor D), como mostra a Figura 4.5(b). Já no algoritmo adaptado o mesmo alvo é coberto por três sensores (os sensores A, C e D). Esse aumento na redundância da cobertura visual também acontece nos alvos de número 2 e 3. A Figura 4.5(c) ilustra o sensor visual B cobrindo esses dois alvos que anteriormente, no algoritmo CPGA, não pertenciam a região de cobertura deste sensor. Assim, no algoritmo adaptado os alvos 2 e 3 são cobertos por quatro sensores enquanto que no algoritmo original são cobertos por apenas três sensores. Isso comprova a importância das adaptações realizadas sobre o algoritmo CPGA para o aumento da cobertura redundante da rede de sensores visuais.

Em seguida foi realizada uma série de testes para avaliar, no geral, o percentual do aumento da cobertura redundante sobre os alvos da rede de sensores quando o algoritmo adaptado, ECPGA, é usado em relação ao algoritmo original, CPGA. Foram feitos 10 testes sobre o mesmo cenário e com a mesma quantidade e configuração dos elementos do teste anterior, variando apenas, de forma aleatória, a posição dos alvos e a orientação inicial dos sensores visuais. A Tabela 4.1 apresenta os dados coletados de cada teste, especificando a quantidade de vezes que cada alvo foi coberto ao executar o algoritmo CPGA e o ECPGA. A Figura 4.6 resume o resultado obtido.

Tabela 4.1: Resultados obtidos nos testes dos algoritmos CPGA e ECPGA.

Quantidade de vezes que cada alvo foi coberto pelos sensores visuais										
	Teste N° 1		Teste N° 2		Teste N° 3		Teste N° 4		Teste N° 5	
	CPGA	ECPGA	CPGA	ECPGA	CPGA	ECPGA	CPGA	ECPGA	CPGA	ECPGA
Alvo 1	1	1	3	3	2	2	2	2	5	3
Alvo 2	2	2	1	1	1	2	1	1	3	2
Alvo 3	2	2	1	1	2	4	3	3	1	1
Alvo 4	3	3	2	2	2	2	1	1	1	2
Alvo 5	2	3	1	1	1	2	1	2	2	2
Alvo 6	2	2	4	4	2	2	1	3	0	1
Alvo 7	3	4	2	3	1	1	2	2	1	1
Alvo 8	1	1	3	3	1	1	1	1	2	2
Alvo 9	2	2	3	3	1	1	1	1	1	1
Alvo 10	3	4	2	2	3	2	2	2	1	2
Total	21	24	22	23	16	19	15	18	17	17
	Teste N° 6		Teste N° 7		Teste N° 8		Teste N° 9		Teste N° 10	
	CPGA	ECPGA	CPGA	ECPGA	CPGA	ECPGA	CPGA	ECPGA	CPGA	ECPGA
Alvo 1	4	4	3	3	2	2	2	4	2	2
Alvo 2	2	2	1	1	2	4	2	2	1	3
Alvo 3	1	1	2	1	2	3	2	2	2	2
Alvo 4	2	2	2	2	4	3	3	3	2	2
Alvo 5	2	3	2	1	1	1	2	3	2	2
Alvo 6	3	4	2	1	3	2	3	3	1	2
Alvo 7	3	2	1	3	3	3	1	1	3	1
Alvo 8	1	1	1	3	2	2	1	1	2	2
Alvo 9	2	3	2	2	0	1	2	2	1	2
Alvo 10	3	6	3	3	1	1	1	2	1	1
Total	23	28	19	20	20	22	19	23	17	19

Pode-se verificar, pelos dados apresentados na Tabela 4.1, que o algoritmo ECPGA não só aumenta a redundância visual sobre vários alvos como também consegue cobrir alvos que não foram cobertos pelo algoritmo CPGA, como pode ser visto nos resultados dos testes N° 5 e N° 8 nos alvos 6 e 9, respectivamente. O campo "Total" apresentado na Tabela 4.1 é a soma da quantidade de vezes que cada alvo foi coberto pelos sensores da rede e representa a cobertura redundante geral da rede de sensores para cada um dos algoritmos de cobertura testados. Com os dados dessa tabela foi calculada a porcentagem do aumento da redundância visual provocada pelo algoritmo adaptado quando comparado com o algoritmo original, CPGA.

O resultado apresentado na Figura 4.6 mostra o aumento que houve na redundância visual sobre os alvos da rede de sensores em cada teste, quando comparado o algoritmo adaptado, ECPGA, com o algoritmo original, CPGA. Percebe-se que apenas no teste de número 5 não houve um aumento na redundância da rede. Além disso, observa-se que o melhor resultado chegou a atingir uma melhora na redundância



Figura 4.6: Aumento percentual na redundância visual da RSVSF quando utilizado o algoritmo adaptado.

visual da rede superior a 20%, no teste de número 6. Tirando uma média desse resultado pode-se concluir que houve uma melhora na redundância visual da rede em torno de 12,7%, quando utilizado o algoritmo adaptado ECPGA em relação ao algoritmo CPGA, para o cenário de testes considerado. Além disso, o algoritmo adaptado é capaz de cobrir alvos que não são cobertos pelo algoritmo CPGA.

O próximo teste analisou o algoritmo quando um número maior de sensores e alvos é utilizado. Nessa simulação foram utilizados 100 sensores visuais sem fio e 80 alvos espalhados aleatoriamente pela região de monitoramento. Esse teste considerou o ângulo de visão dos sensores visuais como sendo de 60°, raio de alcance de 30 metros e a região de monitoramento de 240m x 240m, outra consideração importante é que todos os 80 alvos foram cobertos ao menos uma vez, ou seja, não ficaram alvos descobertos nesse teste. A Figura 4.7 apresenta uma comparação, entre os algoritmos CPGA e ECPGA, mostrando a quantidade total de vezes que todos os 80 alvos foram cobertos em cada um dos 10 testes realizados.

Nota-se que em todos os testes houve uma melhoria na cobertura redundante quando o algoritmo adaptado é utilizado frente o algoritmo original CPGA. O resultado mais favorável foi encontrado no teste de número 6 onde o algoritmo ECPGA conseguiu cobrir 216 vezes todos os 80 alvos da aplicação contra 174 vezes obtido pelo algoritmo CPGA. Isso significa um aumento da cobertura visual redundante sobre os alvos de 24,13%. Na média, a melhoria na redundância visual obtida foi de 14,23% mesmo quando o algoritmo é colocado a prova em um cenário mais complexo, devido a maior quantidade de alvos e sensores envolvidos.

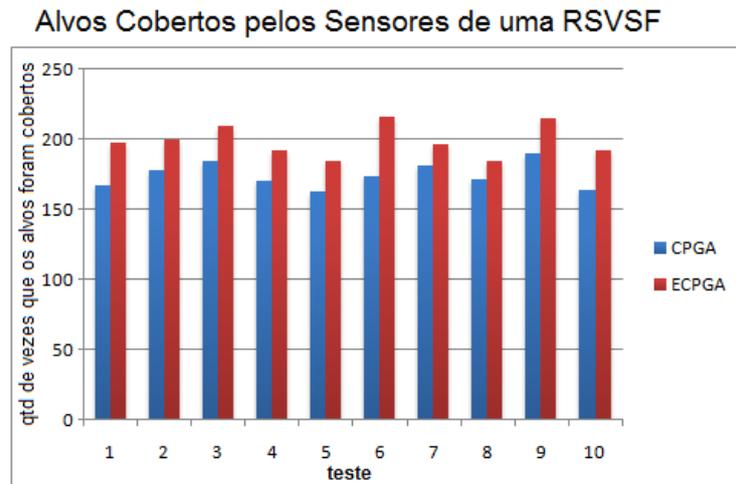


Figura 4.7: Comparação da redundância visual da RSVSF entre o algoritmo CPGA e o ECPGA.

4.2 RCMA: Algoritmo de Maximização da Cobertura Redundante

Foi desenvolvido ainda um novo algoritmo para a maximização da cobertura redundante em redes de sensores visuais sem fio. Esse novo algoritmo foi desenvolvido usando uma estratégia diferenciada em relação aos algoritmos já encontrados na literatura. Enquanto que a literatura propõe algoritmos que buscam a melhor orientação para cada sensor visual de forma a cobrir o máximo número de alvos possíveis, esse novo algoritmo tenta primeiramente resolver o problema da cobertura total dos nós, mesmo que para isso seja necessário que determinados sensores cubram menos alvos do que ele é capaz de cobrir, para posteriormente otimizar a redundância da cobertura visual.

O novo algoritmo desenvolvido foi nomeado de RCMA (Redundant Coverage Maximization Algorithm) e seu funcionamento consiste em quatro passos principais. Essa nova abordagem propõe um método alternativo de tratar a questão da maximização da cobertura redundante de forma que os problemas de cobertura total dos alvos da rede sejam primeiramente resolvidos para posteriormente buscar utilizar os demais sensores visuais para maximizar a redundância visual. O algoritmo é do tipo centralizado, onde as informações são reunidas em um nó central que fica responsável por processá-las e tomar a decisão de melhor orientação para cada sensor.

A descrição dos quatro passos no qual o algoritmo foi dividido permitirá um melhor entendimento de seu funcionamento e suas principais características:

- **Passo 1:** *Cada sensor verifica quais alvos eles podem cobrir.*
- **Passo 2:** *Sensores são ajustados para cobrir os alvos que somente eles podem*

cobrir.

- **Passo 3:** *O algoritmo faz com que todos os alvos sejam cobertos ao menos uma vez.*
- **Passo 4:** *Utiliza-se os sensores que não foram configurados nos passos anteriores para maximizar a cobertura redundante da rede.*

O primeiro passo do algoritmo consiste em verificar quais alvos cada sensor visual da rede pode cobrir, em cada orientação possível de se ajustar. Dessa forma, uma estrutura de dados é montada com essas informações de forma que um dispositivo central possa processá-las e tomar a melhor decisão de orientação para cada sensor visual. Um fator decisivo para o correto funcionamento desse algoritmo é manter as informações dessa estrutura de dados consistente a todo o momento. Nesse primeiro passo, foi reutilizado o conceito, desenvolvido anteriormente no algoritmo ECPGA, de criação de setores com intersecções entre as regiões de cobertura nos quais as câmeras podem se ajustar, como pode ser visto na Figura 4.8.

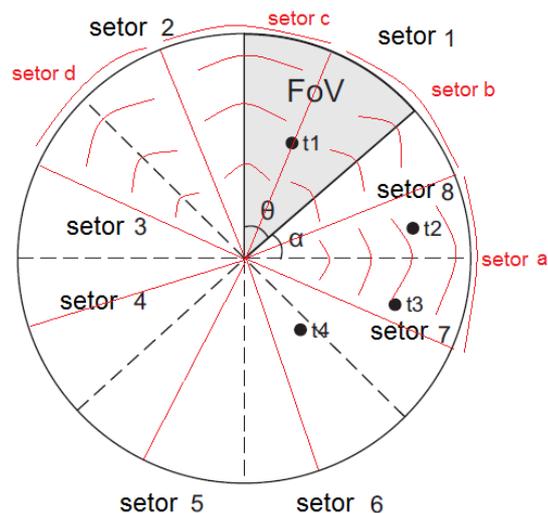


Figura 4.8: Criação de setores com intersecções para ajuste da orientação do FoV dos sensores visuais.

O segundo passo do algoritmo tem por objetivo cobrir todos os alvos que só podem ser cobertos por apenas um sensor visual. Assim, se existirem alvos que só podem ser cobertos por determinados sensores, estes sensores irão ajustar sua orientação para cobri-los de forma a evitar que estes alvos fiquem descobertos. Essa abordagem busca evitar que alvos possam ficar sem cobertura como acontecia no uso de algoritmos gulosos. Em muitos casos, os sensores configurados nesse segundo passo não ajustam sua orientação na posição que irá cobrir a maior quantidade de alvos, mas numa posição que evitará que alvos fiquem descobertos. A Figura 4.9(a) apresenta a configuração inicial da rede de sensores visuais sem fio e a Figura 4.9(b) apresenta o resultado obtido especificamente pelo segundo passo do algoritmo.

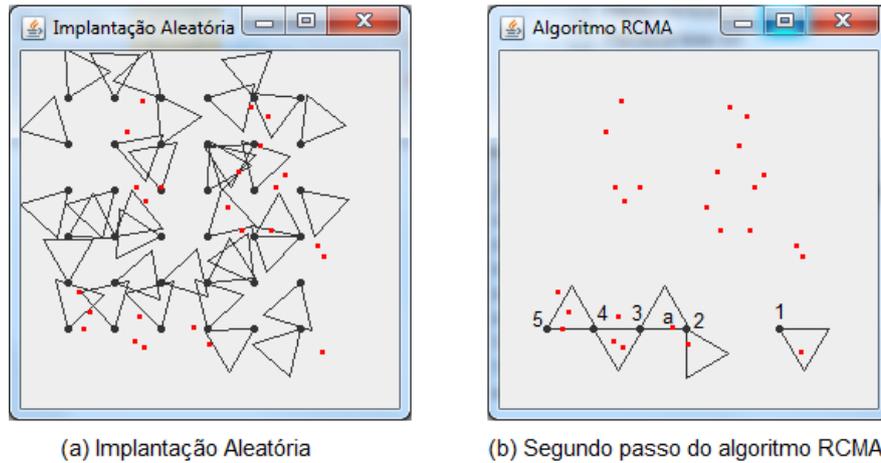


Figura 4.9: Resultados iniciais obtidos com a execução do algoritmo RCMA.

Na Figura 4.9(b), os alvos cobertos pelos sensores ativos no cenário são os alvos que só podem ser cobertos por um sensor. É importante notar que, quando um sensor fixa seu FoV em determinada orientação, os demais alvos que este sensor poderia cobrir em outras orientações não estarão mais disponíveis. Assim, essa informação precisa ser atualizada na estrutura de dados e esses alvos devem considerar que a nova quantidade de sensores que ainda podem o cobrir passa a ser de $n - 1$, onde n é a quantidade inicial de sensores pelo qual determinado alvo poderia ser coberto.

Dessa forma, analisando a numeração atribuída aos sensores ativos na Figura 4.9(b), percebe-se que os sensores visuais 1 e 2 fazem a cobertura dos alvos que somente eles poderiam de fato cobrir, já que não existe outro sensor no cenário que possa cobrir esses alvos. Quando o sensor 2 fixa sua orientação, o alvo "a", que inicialmente poderia ser coberto tanto pelo sensor 2 quanto pelo sensor 3, agora somente pode ser coberto pelo sensor 3. Ou seja, após a configuração do sensor 2 o alvo "a" só pode ser coberto por apenas um sensor (o sensor 3) e é por causa disso que ele foi coberto ainda no segundo passo do algoritmo. A mesma situação acontece com os alvos cobertos pelos sensores 4 e 5. Portanto, nota-se a importância de realizar atualizações nas informações da estrutura de dados a cada evento que ocorre.

O terceiro passo do algoritmo RCMA consiste em cobrir, pelo menos uma vez, os alvos que ainda não foram cobertos. Ao final desse passo, todos os alvos, que podem ser cobertos, deverão estar cobertos por algum sensor visual. Com isso, obteremos uma configuração parcial da rede na qual todos os alvos estarão cobertos e, em muitos dos casos, usando a menor quantidade de sensores possível. A Figura 4.10 apresenta o resultado obtido em cada etapa do algoritmo RCMA e a Figura 4.10(c) apresenta o resultado obtido especificamente no terceiro passo desse novo algoritmo.

Nesse terceiro passo, exceto o caso em que nenhum sensor pode cobrir determinado alvo, para cada alvo ainda não coberto haverá mais de um sensor visual que pode cobri-lo. Assim, na implementação desse algoritmo definiu-se que o sensor escolhido,

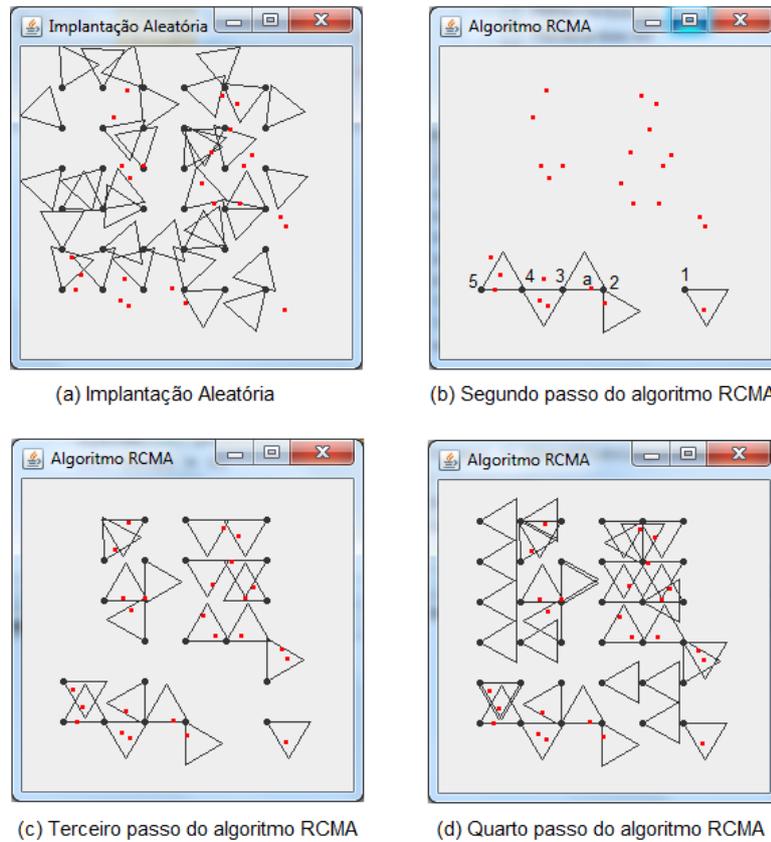


Figura 4.10: Resultados obtidos em cada passo da execução do algoritmo RCMA.

dentre os possíveis para realizar a cobertura de um alvo, seria o sensor que, no total (incluindo todas as orientações), cobrisse a menor quantidade de alvos. Com isso, os demais sensores disponíveis, que podem cobrir mais alvos, serão preservados para que no próximo passo possam ser utilizados para maximizar a redundância, como comprovado nos testes realizados.

O quarto e último passo do algoritmo tem o objetivo de utilizar os sensores visuais que ainda não foram configurados para maximizar a cobertura redundante da rede de sensores visuais sem fio. Assim, cada sensor é posicionado em uma determinada orientação de forma a cobrir o máximo de alvos possíveis. Nesse passo do algoritmo, foi agregado ainda o conceito de prioridade de alvos proposto no algoritmo CPGA desenvolvido por [Costa et al. 2014b]. Com isso, quando houver alvos com diferentes prioridades no cenário, os sensores visuais irão, sempre que possível, ajustar sua orientação para cobrir os alvos com maior prioridade. A Figura 4.10(d) apresenta o resultado obtido após o quarto passo do algoritmo RCMA.

O Algoritmo 4 apresenta o pseudocódigo que realiza o primeiro passo do RCMA. Ele basicamente verifica quais alvos que um sensor pode cobrir em cada orientação possível do sensor ser ajustado. Conforme a verificação vai sendo realizada, informações são coletadas para montar uma estrutura de dados que dará

suporte ao funcionamento dos demais passos do RCMA. As informações armazenadas na estrutura de dados, através dos métodos *addViewerSensor(sensor)* e *addDataTarget(target, orientation)*, permitem responder a questões como: Quais sensores podem cobrir determinado alvo e em qual orientação? Quantos sensores podem cobrir determinado alvo? E quais alvos podem ser cobertos por determinado sensor? A variável *Targets* é uma lista de todos os alvos da aplicação, a variável *Sensors* é uma lista com todos os sensores da rede de sensores e o método *TVT(orientation, target)* é o mesmo usado no algoritmo ECPGA.

Algorithm 4 Pseudocódigo do primeiro passo do algoritmo RCMA

```

1: procedure firstStep(Sensors, Targets)
2:   unusedSensors[]
3:   for  $i \leftarrow 1$  to Sensors.size() do
4:     visualSensor  $\leftarrow$  Sensors.get(i)
5:      $\theta \leftarrow$  visualSensor.getAngle()
6:     TotalOrientation  $\leftarrow 2\pi/(\theta/2)$ 
7:     for  $j \leftarrow 1$  to TotalOrientation do
8:       for  $t \leftarrow 1$  to Targets.size() do
9:         target  $\leftarrow$  Targets.get(t)
10:        if visualSensor.TVT((j *  $\theta/2$ )mod(2 $\pi$ ), target) then
11:          target.addViewerSensor(visualSensor)
12:          visualSensor.addDataTarget(target, (j *  $\theta/2$ )mod(2 $\pi$ ))
13:        end if
14:      end for
15:    end for
16:    unusedSensors.add(visualSensor)
17:  end for
18:  secondStep(unusedSensors, Targets)
19: end procedure

```

O Algoritmo 5 apresenta o segundo passo do RCMA. Essa etapa tem o objetivo de cobrir os alvos que só podem ser cobertos por um sensor. Após a execução do primeiro passo do RCMA sobre cada sensor visual da rede, as informações coletadas foram associadas aos sensores e uma lista de sensores visuais foi criada com a estrutura de dados montada. Essa lista é representada pela variável *unusedSensors* e é utilizada nos demais passos do algoritmo RCMA, como pode ser visto no Algoritmo 5 e no Algoritmo 6.

Algorithm 5 Pseudocódigo do segundo passo do algoritmo RCMA

```

1: procedure secondStep(UNUSEDSENSORS, TARGETS)
2:   configured[]
3:   controle ← TRUE
4:   while controle do
5:     controle ← FALSE
6:     for t ← 1 to TARGETS.size() do
7:       targ ← TARGETS.get(t)
8:       ViewerSensors[] ← targ.getViewerSensor()
9:       if ViewerSensors.size() == 1 AND targ.isNotViewed() then
10:        controle ← TRUE
11:        vSensor ← ViewerSensors.get(1)
12:        dataTargets[] ← vSensor.getDataTargets()
13:        for a ← 1 to dataTargets.size() do
14:          dataT ← dataTargets.get(a)
15:          CoveredTarget ← dataT.getCoveredTarget()
16:          if CoveredTarget == targ then
17:            newOrientation ← dataT.getSensorOrientation()
18:            vSensor.setOrientation(newOrientation)
19:          end if
20:        end for
21:        vSensor.updateCoveredTargets()
22:        configured.add(vSensor)
23:        UNUSEDSENSORS.remove(vSensor)
24:      end if
25:    end for
26:  end while
27:  thirdStep(UNUSEDSENSORS, configured, TARGETS)
28: end procedure

```

O método *updateCoveredTargets()*, usado tanto no Algoritmo 5 quanto no Algoritmo 6, é responsável em atualizar a estrutura de dados sinalizando quais alvos passaram ser cobertos e quais alvos não poderão mais ser cobertos por determinado sensor após fixar sua orientação. Além disso, a variável *configured* é a lista de sensores visuais que já foram configurados, ou seja, sua orientação já foi definida e não será mais alterada.

O terceiro passo do RCMA é apresentado em pseudocódigo no Algoritmo 6. O código é similar ao apresentado no segundo passo exceto pelo fato desse algoritmo trabalhar na cobertura dos alvos que só podem ser cobertos por dois ou mais sensores. O trecho de código compreendido entre as linhas 12 e 19 do Algoritmo 6 é responsável em verificar, na lista de sensores que podem cobrir determinado alvo, qual sensor cobre menos alvos, considerando todas as orientações. Dessa forma, os sensores que cobrirem menos alvos serão os selecionados para realizar a cobertura dos alvos ainda não cobertos. No final do Algoritmo 6 irá sobrar apenas os sensores que ainda não foram configurados mas que têm o potencial de cobrir a maior quantidade de alvos. Esses sensores serão utilizados no último passo do algoritmo RCMA para maximizar a redundância da cobertura visual.

Algorithm 6 Pseudocódigo do terceiro passo do algoritmo RCMA

```

1: procedure thirdStep(UNUSEDSENSORS, CONFIGURED, TARGETS)
2:   controle ← TRUE
3:   while controle do
4:     controle ← FALSE
5:     for t ← 1 to TARGETS.size() do
6:       targ ← TARGETS.get(t)
7:       VIEWERSENSORS[] ← targ.getViewerSensor()
8:       if VIEWERSENSORS.size() > 1 AND targ.isNotViewed() then
9:         controle ← TRUE
10:        vSensor
11:        minTargetsCovered ← 99999
12:        for x ← 1 to VIEWERSENSORS.size() do
13:          auxSensor ← VIEWERSENSORS.get(x)
14:          auxTargetsCovered ← auxSensor.getTargetCovered()
15:          if auxTargetsCovered.size() < minTargetsCovered then
16:            minTargetsCovered ← auxTargetsCovered.size()
17:            vSensor ← auxSensor
18:          end if
19:        end for
20:        dataTargets[] ← vSensor.getDataTargets()
21:        for a ← 1 to dataTargets.size() do
22:          dataT ← dataTargets.get(a)
23:          CoveredTarget ← dataT.getCoveredTarget()
24:          if CoveredTarget == targ then
25:            newOrientation ← dataT.getSensorOrientation()
26:            vSensor.setOrientation(newOrientation)
27:          end if
28:        end for
29:        vSensor.updateCoveredTargets()
30:        CONFIGURED.add(vSensor)
31:        UNUSEDSENSORS.remove(vSensor)
32:      end if
33:    end for
34:  end while
35:  fourthStep(UNUSEDSENSORS, CONFIGURED, TARGETS)
36: end procedure

```

O último passo do algoritmo RCMA é realizado, basicamente, utilizando o método $checkViewing(v, Targets)$ desenvolvido no ECPGA e que pode ser visto no Algoritmo 1. Esse método foi reutilizado porque realiza basicamente o que o quarto passo do algoritmo RCMA propõe: o ajuste da orientação dos sensores na posição que cobre a maior quantidade de alvos levando em consideração a sua prioridade. Porém, no algoritmo RCMA, o método $checkViewing(v, Targets)$ não é usado em todos os sensores mas apenas sobre aqueles nós que até então não tinham sido configurados nos passos anteriores, ou seja, os sensores que sobraram da variável $unusedSensors$. O objetivo desse último passo é utilizar os sensores restantes para maximizar a cobertura redundante sobre os alvos já que todos já foram cobertos ao menos uma vez.

4.2.1 Testes e Comparações

O primeiro teste realizado buscou avaliar o funcionamento do algoritmo RCMA. Para isso, o cenário considerado consiste em 36 sensores visuais e 25 alvos numa região de 210m x 210m. Os sensores visuais foram configurados com ângulo de visão

de 60° e raio de alcance de 30 metros. A orientação dos sensores visuais bem como a distribuição dos alvos ao longo do cenário são determinadas aleatoriamente. Por fim, a Figura 4.11 apresenta os resultados parciais e o final obtidos com o teste do algoritmo RCMA.

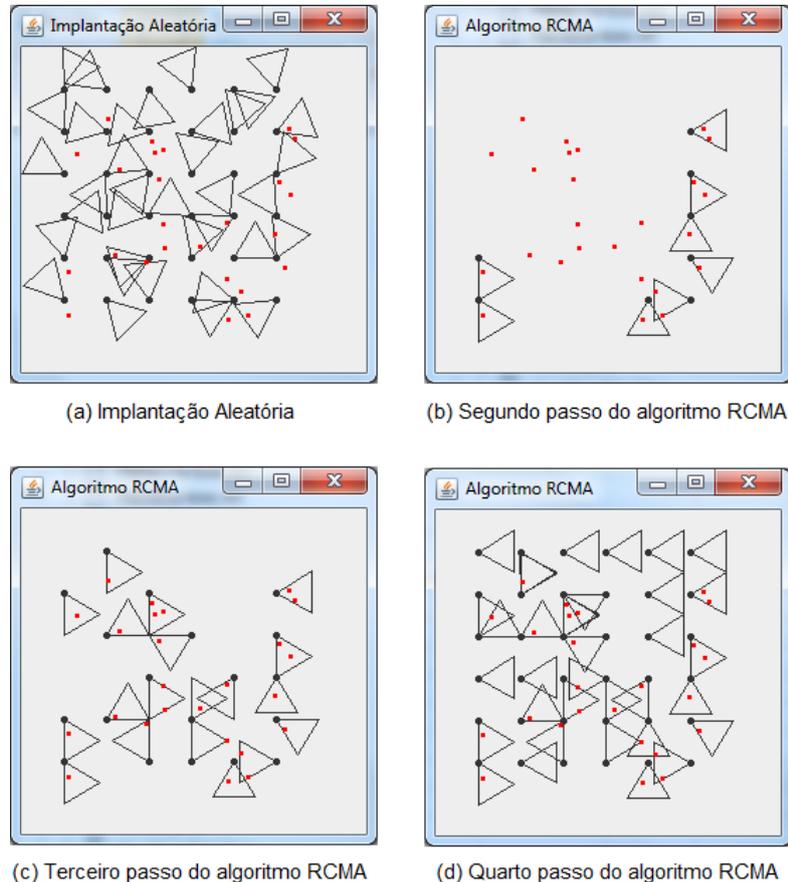


Figura 4.11: Resultados obtidos em cada passo do teste do algoritmo RCMA.

A Figura 4.11(a) mostra a configuração inicial dos sensores, sendo esse o ponto de partida do algoritmo RCMA. A Figura 4.11(b) apresenta a situação do cenário após a execução do segundo passo do algoritmo, quando é realizada a cobertura daqueles alvos que só podem ser cobertos por apenas um sensor. Nesse exemplo, são 11 alvos que são cobertos por um total de 8 sensores. O resultado do terceiro passo do algoritmo é apresentado na Figura 4.11(c). Nesse passo, todos os alvos que ainda não foram cobertos passam a ser cobertos por uma quantidade reduzida de sensores. Nesse caso, apenas 19 sensores bem ajustados foram suficientes para cobrir todos os 25 alvos da rede de sensores visuais. A Figura 4.11(d) apresenta o resultado final do algoritmo RCMA, após o quarto e último passo desse algoritmo os sensores visuais estão configurados de forma a obter a máxima cobertura redundante dos alvos. Esse quarto passo tem o objetivo específico de usar os sensores ainda não configurados para maximizar a cobertura redundante sobre os alvos da aplicação. Nesse exemplo,

após o quarto passo, os sensores visuais da rede cobrem todos os 25 alvos por 36 vezes, sendo este um resultado muito superior ao obtido na implantação aleatória da Figura 4.11(a), que cobre apenas 7 alvos sem nenhuma redundância.

No próximo teste é feita uma comparação entre os algoritmos CPGA e os já desenvolvidos (ECPGA e RCMA) quando testados sob o mesmo cenário. O cenário de teste consiste em 16 sensores e 10 alvos numa região de 120m x 120m, sendo que os sensores visuais têm ângulo de visão de 60° e raio de alcance de 30 metros. A Figura 4.12 apresenta o resultado obtido em cada algoritmo. O algoritmo CPGA desenvolvido por [Costa et al. 2014b] cobriu todos os 10 alvos por 16 vezes enquanto que o algoritmo adaptado, ECPGA, conseguiu cobrir todos os alvos por 17 vezes. Porém, o algoritmo que obteve melhor resultado foi o novo algoritmo RCMA, que conseguiu cobrir todos os alvos por 21 vezes, proporcionando, assim, a melhor cobertura redundante de alvos para a rede de sensores visuais sem fio.

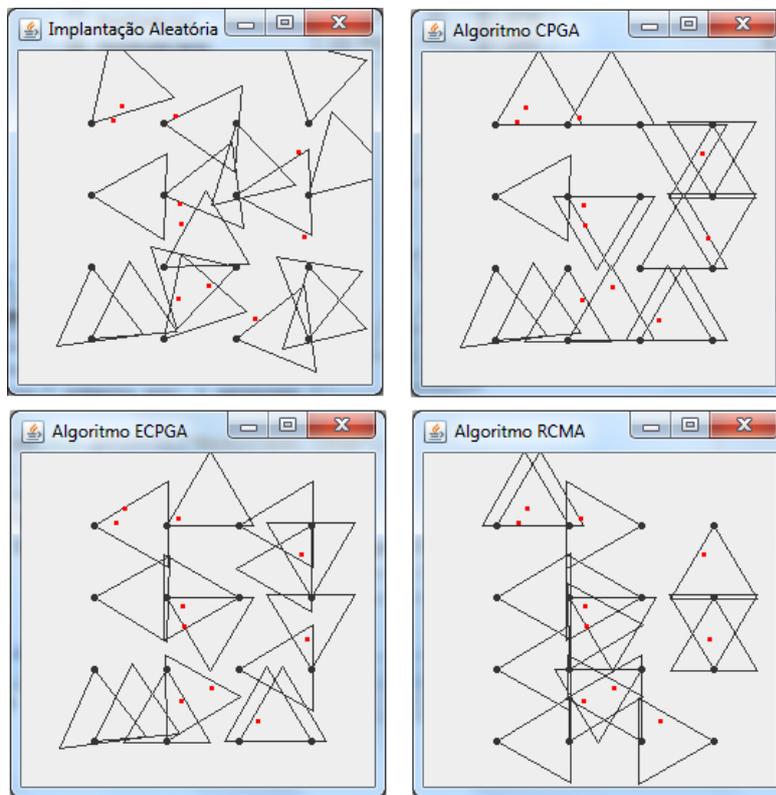


Figura 4.12: Resultado obtido em cada algoritmo.

Para avaliar o ganho de desempenho do algoritmo RCMA frente aos demais algoritmos, uma série de 10 testes foi realizada em um cenário mais complexo, envolvendo 80 alvos e 100 sensores com ângulo de visão de 60° e raio de alcance de 30 metros. Em cada teste uma nova posição aleatória era atribuída aos alvos e às orientações dos sensores visuais. O objetivo é avaliar a real melhoria na cobertura redundante dos alvos obtida pelo novo algoritmo quando comparado com os demais algoritmos

já testados. A Figura 4.13 apresenta um gráfico resumindo o resultado obtido por cada algoritmo em cada um dos 10 testes realizados. É importante ressaltar que em todos os testes, e por todos os algoritmos, houve a cobertura de todos os 80 alvos do cenário.

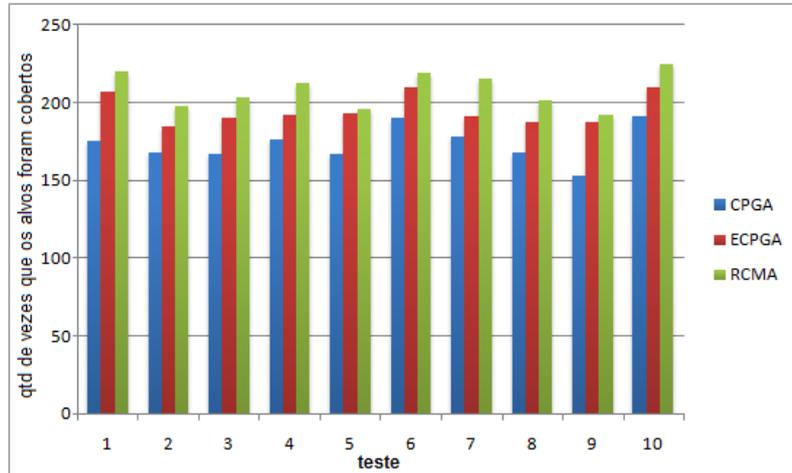


Figura 4.13: Comparação entre os algoritmos CPGA, ECPGA e RCMA.

Percebe-se, pela Figura 4.13, que em todos os testes o desempenho do algoritmo RCMA foi superior tanto em relação ao algoritmo CPGA quanto em relação ao ECPGA. O RCMA foi o algoritmo que ofereceu a maior cobertura redundante de alvos em cada teste, sendo que o melhor desempenho foi obtido no teste de número 1 onde obteve um aumento na redundância de 25,7% em relação ao algoritmo CPGA, enquanto que o desempenho do ECPGA foi de 18,3% quando também comparado com o algoritmo CPGA. Na média, o algoritmo adaptado ECPGA apresentou um aumento na redundância de 12,81% quando comparado com algoritmo CPGA. Já o algoritmo RCMA obteve um aumento médio de 20,25% na cobertura redundante de alvos quando comparado também com relação o CPGA. Dessa forma, pode-se constatar que o algoritmo RCMA além de apresentar bons resultados também é melhor que o algoritmo adaptado ECPGA para o aumento da cobertura redundante de alvos nas redes de sensores visuais sem fio.

A Figura 4.14 apresenta a rede de sensores visuais configurada para cada um dos algoritmos avaliados em um dos 10 testes realizados. Pode-se verificar que para os três algoritmos testados (CPGA, ECPGA e RCMA) todos os 80 alvos são cobertos, sendo a diferença de desempenho obtida em relação a quantidade de alvos que são cobertos de forma redundante, que é contabilizado pela quantidade de vezes que todos os alvos foram cobertos por cada algoritmo.

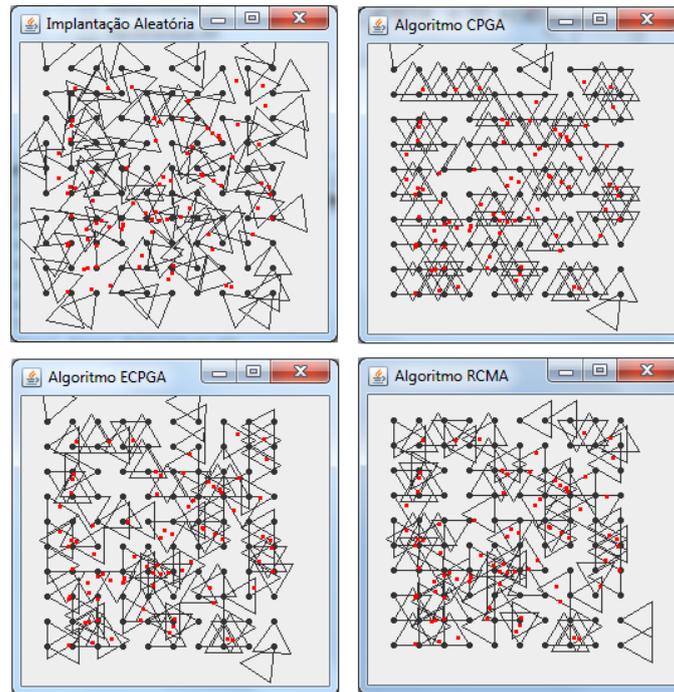


Figura 4.14: A configuração da RSVSF para cada algoritmo em um dos testes comparativos de desempenho.

4.3 EDPGA: Um Algoritmo Distribuído

Os algoritmos ECPGA e RCMA são algoritmos centralizados onde as decisões são tomadas por um dispositivo central e, em seguida, são enviadas a cada sensor da rede as informações sobre a orientação que deve ser assumida. Como mais uma contribuição desse trabalho, foi desenvolvido um algoritmo distribuído, chamado de EDPGA (Enhanced Distributed Prioritized Greedy Algorithm), que tomou como base para sua construção o algoritmo ECPGA. Nesse algoritmo, cada nó deve tomar sua própria decisão sobre a melhor orientação a ser adotada. Para isso, cada sensor da rede precisará trocar dados com os sensores mais próximos em busca de informações necessárias para decidir a sua melhor orientação.

O desenvolvimento de um algoritmo distribuído com o objetivo de aumentar a redundância visual de alvos em uma rede de sensores visuais, buscando ao mesmo tempo a total cobertura dos alvos, é uma abordagem inovadora e não encontrada ainda na literatura. Com isso, pretende-se avaliar o desempenho dessa nova abordagem e compará-la com os algoritmos centralizados já desenvolvidos a fim de verificar suas reais vantagens e desvantagens. O EDPGA é uma versão distribuída do algoritmo ECPGA, ou seja, utiliza basicamente dos mesmos conceitos que o ECPGA, sendo, portanto, um algoritmo guloso e que leva em consideração a prioridade dos alvos da rede de sensores visuais sem fio. A Figura 4.15 ilustra o funcionamento do algoritmo passo a passo.

O algoritmo EDPGA inicia sua execução fazendo com que o sensor visual reconheça todos os alvos que ele pode cobrir nas possíveis orientações de ajuste, conforme a mesma metodologia adotada nos algoritmos anteriores. Em seguida, o sensor visual ajusta sua orientação para a posição na qual ele pode cobrir a maior quantidade de alvos, como mostra a Figura 4.15(b). Se houver duas ou mais orientações que cobrem a mesma quantidade de alvos, então os alvos com maior prioridade serão cobertos. Por se tratar de um algoritmo distribuído, esse mesmo procedimento é realizado por cada sensor da rede de forma individual.

É importante ressaltar que esse ajuste de orientação do algoritmo EDPGA difere do algoritmo ECPGA. O algoritmo ECPGA desenvolvido anteriormente fazia com que cada sensor ajustasse sua orientação para cobrir a maior quantidade de alvos ainda não cobertos, enquanto que o algoritmo EDPGA ajusta os sensores para cobrir a maior quantidade de alvos possível, independente de já estarem sendo cobertos por outros sensores ou não. Isso acontece porque no algoritmo distribuído não é possível fazer com que cada sensor saiba quais alvos estão ou não cobertos sem primeiro trocar informações com os demais sensores da rede.

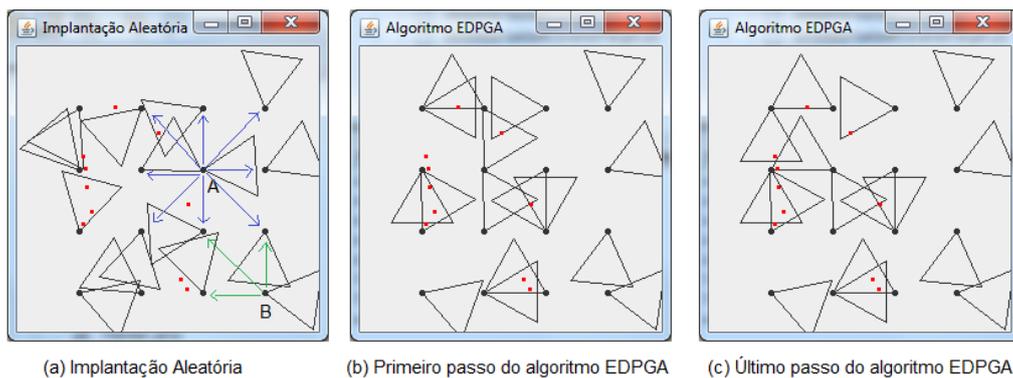


Figura 4.15: Funcionamento do algoritmo EDPGA.

Em seguida, o algoritmo EDPGA faz com que cada sensor encaminhe para os seus vizinhos (os sensores mais próximos) mensagens com informações necessárias para a escolha da melhor orientação. Todos os sensores da rede enviam e recebem essas informações dos seus respectivos vizinhos. Dessa forma, cada sensor visual pode criar uma estrutura de dados que permita analisar quais alvos já estão sendo cobertos pelos seus vizinhos. Assim, cada sensor pode verificar se há alvos que ele pode cobrir, porém que ainda não estão sendo cobertos por nenhum outro sensor. É importante notar que para um sensor realizar a comunicação entre os seus vizinhos há a necessidade de protocolos de comunicação, definição das tecnologias a serem usadas e determinação da taxa de transferência de dados, entre outros detalhes que, em nosso caso, não foi definido, pois os testes foram realizados apenas em ambientes de simulação.

A Figura 4.15(a) exemplifica através de setas azuis e verdes quais são os vizinhos, nesse cenário, dos sensores nomeados de A e B, respectivamente. Os vizinhos de

cada sensor visual, como ilustrado, foram escolhidos de forma determinística selecionando os alvos mais próximos (adjacentes) em todas as direções. Ou seja, não foi desenvolvido nenhum algoritmo para detecção de vizinhos, eles foram escolhidos de forma estática e para funcionar em um cenário onde os sensores fossem implantados igualmente espaçados no cenário conforme apresentado na Figura 4.15(a).

A forma como os vizinhos foram determinados nesse trabalho e a comunicação entre eles é apenas uma simulação do que é esperado na prática quando o algoritmo for testado em um ambiente real. Logicamente, em uma rede real, as mensagens trocadas entre os sensores são enviadas em *broadcast* pelo meio sem fio. Assim, os vizinhos de um determinado sensor visual serão todos os sensores que estão dentro do alcance da sua cobertura de comunicação sem fio. Ou seja, os sensores que estão próximos suficientes para receber a mensagem que foi transmitida pelo meio sem fio serão os vizinhos do sensor que transmitiu a mensagem. Dessa forma, mais ou menos sensores podem ser vizinhos de outro sensor na rede a depender da distribuição dos sensores pela região de monitoramento e do raio de cobertura das antenas de comunicação sem fio desses sensores.

O próximo passo consiste em fazer com que cada sensor analise a estrutura de dados montada no passo anterior e mude sua orientação caso haja alvos descobertos que eles podem cobrir, porém, sem afetar a cobertura total da rede, como pode ser visto na Figura 4.15(c). Se houver mais de uma orientação com alvos que só um sensor pode cobrir, o sensor irá ajustar sua posição na orientação que tiver a maior quantidade de alvos para cobrir.

A cada mudança de orientação realizada por um sensor, novas informações são encaminhadas em uma nova mensagem para os vizinhos manterem a rede atualizada. Quando um vizinho envia uma mensagem nova para outro sensor, esse sensor substitui as informações antigas que tinha armazenado desse vizinho pelas as novas informações recebidas e em seguida verifica novamente se é necessário realizar algum ajuste em sua orientação. O algoritmo finaliza o ajuste de orientação dos sensores quando cada sensor, após receber mensagens de seus vizinhos, verifica que não há mais o que ser mudado em sua orientação e, portanto, não precisa enviar mais nenhuma mensagem de atualização na rede.

O Algoritmo 7 apresenta o procedimento que deve ser executado por cada um dos sensores visuais após sua inicialização no cenário da aplicação. Esse código faz com que cada sensor reconheça todos os alvos que ele pode cobrir, em cada possível orientação de ajuste, e mantém a sua orientação final configurada na posição que cobre a maior quantidade de alvos.

Algorithm 7 Pseudocódigo do EDPGA que determina a orientação inicial dos sensores

```

1: procedure checkViewing()
2:  $\theta \leftarrow this.getAngle()$ 
3:  $TotalOrientation \leftarrow 2\pi/(\theta/2)$ 
4:  $totalViewed \leftarrow 0$ 
5:  $totalPriorities \leftarrow 0$ 
6:  $bestOrientation$ 
7: for  $j \leftarrow 1$  to  $TotalOrientation$  do
8:    $priorities \leftarrow 0$ 
9:    $viewed \leftarrow 0$ 
10:   $Targets \leftarrow this.getTargetsViewed((j * \theta/2) \bmod(2\pi))$ 
11:  for  $t \leftarrow 1$  to  $Targets.size()$  do
12:     $target \leftarrow Targets.get(t)$ 
13:     $viewed \leftarrow viewed + 1$ 
14:     $priorities \leftarrow priorities + target.getPt()$ 
15:  end for
16:  if  $viewed > totalViewed$  then
17:     $totalViewed \leftarrow viewed$ 
18:     $bestOrientation \leftarrow (j * \theta/2) \bmod(2\pi)$ 
19:     $totalPriorities \leftarrow priorities$ 
20:  else
21:    if  $viewed == totalViewed$  then
22:      if  $priorities > totalPriorities$  then
23:         $totalPriorities \leftarrow priorities$ 
24:         $bestOrientation \leftarrow (j * \theta/2) \bmod(2\pi)$ 
25:      end if
26:    end if
27:  end if
28: end for
29:  $this.setOrientation(bestOrientation)$ 
30:  $TargetsCovered[] \leftarrow this.getTargetsViewed(bestOrientation)$ 
31:  $this.send(TargetsCovered)$ 
32: end procedure

```

O método $getTargetsViewed(orientation)$ é responsável em retornar uma lista de alvos que podem ser cobertos pelo sensor em determina orientação. Para o ambiente de simulação esse método é similar ao *Target Viewing Test (TVT)* usado no algoritmo CPGA, para a realização de testes em um ambiente real é necessário o uso de técnicas de computação visual para o reconhecimento dos alvos. No final do Algoritmo 7, através do método $send(TargetsCovered)$, as informações dos alvos cobertos são enviadas pela rede para que os sensores vizinhos possam usar essas informações na próxima etapa do algoritmo EDPGA.

O próximo passo da execução do EDPGA é apresentado no algoritmo 8. Essa etapa é executada após os sensores visuais receberem as mensagens de atualização de seus vizinhos e consiste em avaliar a necessidade de alteração em sua orientação e, caso haja mudança de orientação, enviam para seus vizinhos uma nova mensagem para informar essas mudanças.

Algorithm 8 Pseudocódigo do algoritmo EDPGA responsável em atualizar a orientação dos sensores visuais

```

1: procedure updateOrientation()
2:  $tNotCoveredNeighbor[] \leftarrow this.getTargetNotCoveredNeighbor()$ 
3: if  $tNotCoveredNeighbor.size() == 1$  then
4:    $target \leftarrow tNotCoveredNeighbor.get(1)$ 
5:    $newOrientation \leftarrow target.getSensorOrientation()$ 
6:    $this.setOrientation(newOrientation)$ 
7: else
8:   if  $tNotCoveredNeighbor.size() > 1$  then
9:      $totalTargetsSameOrient \leftarrow 0$ 
10:     $bestOrientat$ 
11:    while  $!tNotCoveredNeighbor.isEmpty()$  do
12:       $nTargetsSameOrient \leftarrow 0$ 
13:       $target \leftarrow tNotCoveredNeighbor.get(1)$ 
14:       $orientat \leftarrow target.getSensorOrientation()$ 
15:       $listAux[]$ 
16:      for  $a \leftarrow 1$  to  $tNotCoveredNeighbor.size()$  do
17:         $targetDiscovered \leftarrow tNotCoveredNeighbor.get(a)$ 
18:         $orientationTD \leftarrow targetDiscovered.getSensorOrientation()$ 
19:        if  $orientationTD == orientat$  then
20:           $nTargetsSameOrient \leftarrow nTargetsSameOrient + 1$ 
21:           $listAux.add(targetDiscovered)$ 
22:        end if
23:      end for
24:       $tNotCoveredNeighbor.removeAll(listAux)$ 
25:      if  $nTargetsSameOrient > contTotal$  then
26:         $bestOrientat \leftarrow orientat$ 
27:         $totalTargetsSameOrient \leftarrow nTargetsSameOrient$ 
28:      end if
29:    end while
30:     $this.setOrientation(bestOrientat)$ 
31:     $TargetsCovered[] \leftarrow this.getTargetsViewed(bestOrientat)$ 
32:     $this.send(TargetsCovered)$ 
33:  end if
34: end if
35: end procedure

```

A variável $tNotCoveredNeighbor$ é uma lista de alvos que estão no alcance de cobertura do sensor visual mas que não estão sendo cobertos pelos seus vizinhos. Essa lista é montada a partir das informações recebidas de todos os vizinhos de determinado sensor. A cada mensagem nova recebida de um vizinho essa lista é atualizada e com base nela que os sensores tomam a decisão de mudar sua orientação ou não. O método $updateOrientation()$ também trata o caso no qual pode haver vários alvos ainda não cobertos em diversas orientações. Nesse caso, a orientação escolhida será a que cobrir a maior quantidade de alvos não cobertos.

Na seção seguinte, os resultados dos testes com o algoritmo distribuído EDPGA serão apresentados bem como comparações mais detalhadas de desempenho dos três algoritmos construídos.

4.3.1 Testes e Comparações

No primeiro teste é feito uma comparação entre os algoritmos CPGA, ECPGA, RCMA e o distribuído, EDPGA, quando aplicados sob o mesmo cenário. O cenário

de teste consiste em 16 sensores e 10 alvos numa região de 150m x 150m, sendo que os sensores visuais têm ângulo de visão de 60° e raio de alcance de 30 metros. A Figura 4.16 apresenta o resultado obtido em cada algoritmo. O algoritmo CPGA desenvolvido por [Costa et al. 2014b] cobriu 9 dos 10 alvos por 15 vezes enquanto que o algoritmo ECPGA e o RCMA conseguiram cobrir todos os alvos por 15 vezes. Já o algoritmo distribuído EDPGA cobriu todos os alvos por 16 vezes sendo, neste exemplo, a maior cobertura redundante de alvos alcançada na rede de sensores visuais sem fio.

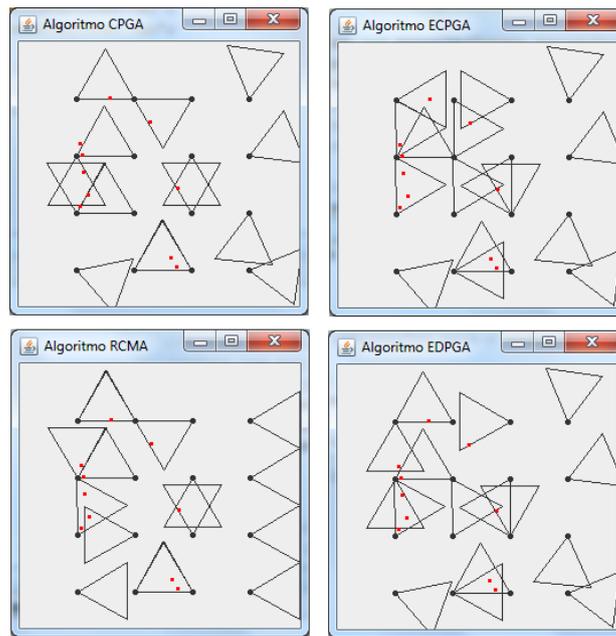


Figura 4.16: Comparação entre o algoritmo distribuído e os algoritmos centralizados.

Assim como feito para os algoritmos anteriores, uma nova série de testes foi realizada para verificar o desempenho do algoritmo distribuído em relação aos demais. O cenário dos testes consiste de 80 alvos e 100 sensores com ângulo de visão de 60° e raio de alcance de 30 metros. Para cada teste uma nova posição aleatória foi atribuída aos alvos e às orientações dos sensores visuais. A Figura 4.17 apresenta o desempenho dos algoritmos em cada teste.

Como pode ser visto na Figura 4.17, o algoritmo EDPGA é o segundo melhor algoritmo para o aumento da redundância visual na rede de sensores visuais sem fio, ficando atrás somente do algoritmo RCMA. Porém, apesar do algoritmo EDPGA apresentar resultados de cobertura redundante superiores aos algoritmos CPGA e ECPGA, em alguns dos testes ele não foi capaz de cobrir todos os 80 alvos do cenário. A Tabela 4.2 apresenta mais detalhes dos testes realizados, inclusive o aumento percentual da cobertura redundante provocado por cada algoritmo desenvolvido quando comparados com o algoritmo CPGA, desenvolvido por [Costa et al. 2014b]. Também é apresentado na Tabela 4.2, a quantidade de alvos cobertos pelos algoritmos

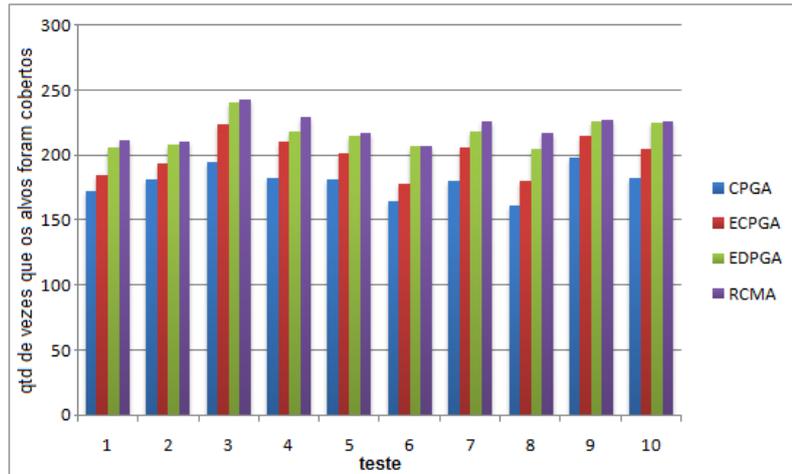


Figura 4.17: Desempenho dos algoritmos desenvolvidos.

em cada teste realizado e por quantas vezes esses alvos foram cobertos pelos sensores da rede.

Tabela 4.2: Resultados obtidos nos testes dos algoritmos centralizados e do EDPGA

Nº Teste	Cobertura dos Alvos (total de alvos cobertos / nº de vezes cobertos)				Aumento na Cobertura Redundante(%)		
	CPGA	ECPGA	EDPGA	RCMA	ECPGA	EDPGA	RCMA
teste 1	79 / 172	80 / 185	79 / 206	80 / 211	7,55 %	19,76 %	22,67 %
teste 2	80 / 181	80 / 194	78 / 208	80 / 210	7,18 %	14,91 %	16,02 %
teste 3	80 / 195	80 / 224	80 / 241	80 / 243	14,87 %	23,58 %	24,61 %
teste 4	80 / 182	80 / 210	80 / 218	80 / 229	15,38 %	19,78 %	25,82 %
teste 5	79 / 181	80 / 201	79 / 215	80 / 217	11,04 %	18,78 %	19,88 %
teste 6	80 / 164	80 / 178	78 / 207	80 / 207	8,53 %	26,21 %	26,2 %
teste 7	79 / 180	80 / 206	78 / 218	80 / 226	14,44 %	21,11 %	25,55 %
teste 8	80 / 161	80 / 180	80 / 205	80 / 217	11,80 %	27,32 %	34,78 %
teste 9	80 / 198	80 / 215	80 / 226	80 / 227	8,58 %	14,14 %	14,64 %
teste 10	80 / 182	80 / 205	80 / 225	80 / 226	12,63 %	23,62 %	24,17 %
				Média	12,20 %	20,90 %	23,44 %

Segundo a Tabela 4.2, o algoritmo EDPGA apresenta uma cobertura redundante mais alta que os algoritmos CPGA e ECPGA em todos os testes, porém, em alguns casos (marcados em vermelho) o algoritmo não consegue cobrir todos os alvos da rede. Assim, apesar do algoritmo distribuído conseguir cobrir a maior parte dos alvos do cenário com uma redundância superior ao de outros algoritmos, ele não é capaz de oferecer uma cobertura total dos alvos em todos os casos.

Além disso, a Tabela 4.2 mostra mais uma vez que o algoritmo RCMA é superior a todos os outros, conseguindo cobrir todos os alvos e com a maior redundância visual. O RCMA chega a alcançar, no melhor dos casos, uma cobertura redundante de 34,78% maior que o algoritmo CPGA. Na média, o algoritmo RCMA apresenta uma cobertura redundante de 23,44% superior em relação ao algoritmo CPGA, 11% em relação ao algoritmo ECPGA e 2% em relação ao algoritmo EDPGA. Outros

percentuais de desempenho entre o algoritmo CPGA e os algoritmos ECPGA e EDPGA podem ser conferidas na Tabela 4.2.

Mais uma série de testes foi realizada, dessa vez para analisar o desempenho dos algoritmos quando o raio de alcance dos sensores visuais é alterado em 20, 30 e 40 metros. O cenário de teste consiste em 50 alvos e 100 sensores visuais e o ângulo de visão dos sensores foi fixado em 60° . Para cada raio de alcance analisado os algoritmos foram testados uma série de 10 vezes e a média foi calculada para obter o resultado apresentado na Figura 4.18. Assim, no total foram realizados 120 testes (3 DoF x 4 Algoritmos x 10 execuções).

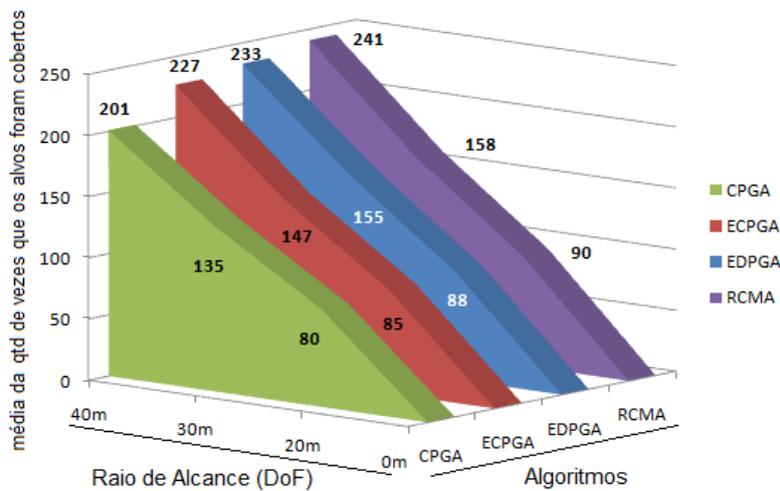


Figura 4.18: Resultado obtido por cada algoritmo variando o raio de alcance dos sensores visuais.

Percebe-se, pela Figura 4.18, que a dimensão do raio de alcance impacta diretamente na cobertura redundante da rede de sensores visuais sem fio e que os resultados obtidos seguem o mesmo padrão de desempenho apresentado nos testes anteriores, com o algoritmo RCMA apresentando os melhores resultados novamente.

Outro conjunto de testes realizado buscou analisar o comportamento dos algoritmos para diferentes ângulos de visão (AoV) em um cenário composto de 50 alvos e 100 sensores visuais. Os algoritmos foram testados configurando o ângulo de visão dos sensores visuais em 30° , 45° e 60° e mantendo o raio de alcance fixo em 30 metros. Para cada ângulo de visão analisado os algoritmos também foram testados uma série de 10 vezes e a média foi calculada para obter o resultado apresentado na Figura 4.19. Ou seja, no total foram realizados mais 120 testes (3 AoVs x 4 Algoritmos x 10 execuções). Além disso, verifica-se que mais uma vez o algoritmo RCMA apresenta o melhor resultado na cobertura redundante em relação aos demais algoritmos.

Por fim, foi realizada uma série de 10 testes variando também a posição dos sensores visuais de forma aleatória, assim como é feito para os alvos. O objetivo é avaliar o

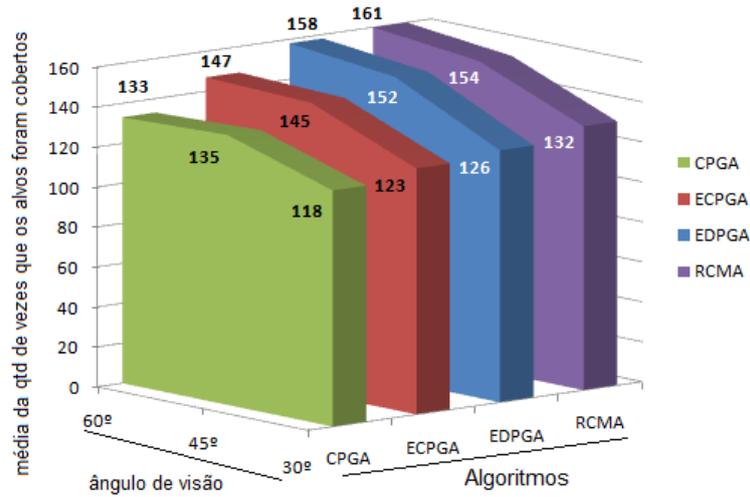


Figura 4.19: Resultado obtido por cada algoritmo variando o ângulo de visão dos sensores visuais.

resultados dos algoritmos em uma situação mais próxima da real, quando os sensores visuais são implantados de forma aleatória. O cenário de teste é composto por 50 alvos e 100 sensores visuais, sendo que os sensores são configurados com ângulo de visão de 60° e raio de alcance de 30 metros. A Figura 4.20 apresenta os resultados obtidos em cada um dos 10 testes realizados.

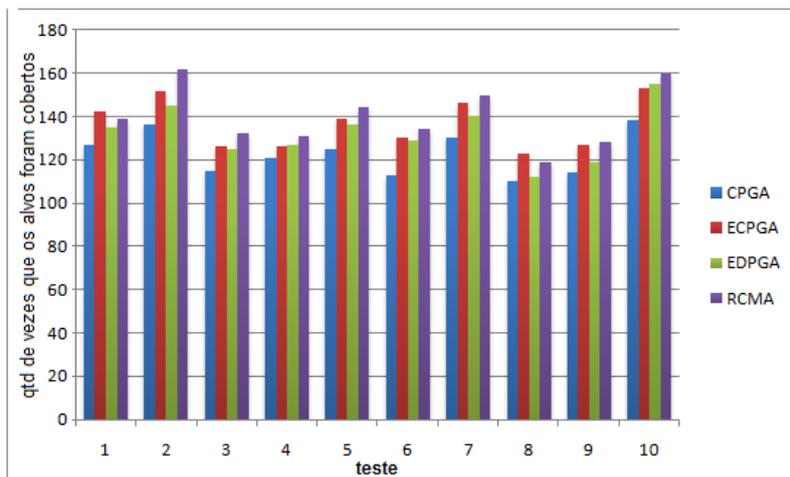


Figura 4.20: Comparação entre os algoritmos quando os sensores visuais são posicionados de forma aleatória.

A Figura 4.20 mostra uma variação significativa nos resultados obtidos em cada teste devido o posicionamento aleatório dos sensores sobre o cenário da aplicação. Em alguns casos, principalmente nos testes que apresentaram os piores desempenhos como os testes de nº 8 e nº 9, alguns alvos nem chegaram a ser cobertos pelos algoritmos de cobertura devido ao posicionamento atribuído aos sensores visuais. Entretanto, o

algoritmo RCMA se manteve, mais uma vez, com os melhores resultados na maioria dos testes realizados.

Na média, o algoritmo EDPGA gerou um aumento na cobertura visual redundante da rede de 7,57% em relação ao algoritmo CPGA. Já o algoritmo ECPGA obteve uma melhora de 11% e o RCMA de 13,7%, ambos também em relação ao algoritmo CPGA. A Figura 4.21 ilustra a configuração obtida pelos sensores visuais no teste de nº 10 para cada algoritmo avaliado.

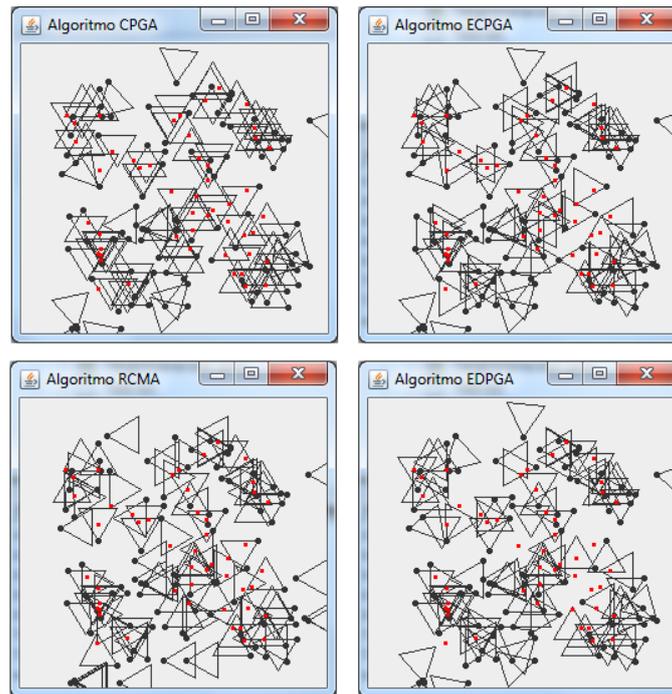


Figura 4.21: Configuração dos sensores visuais após a execução de cada algoritmo no teste de nº 10.

Na Figura 4.21 é possível visualizar alguns alvos que não foram cobertos após a execução do algoritmo EDPGA, problema esse já apresentado na tabela 4.2. Nesse caso, 7 dos 50 alvos não foram cobertos o que mostra que o posicionamento aleatório dos alvos impactou mais ainda o desempenho desse algoritmo distribuído. Além disso, percebe-se que o algoritmo EDPGA, em 8 dos 10 testes, apresentou uma perda do padrão de desempenho em relação o algoritmo ECPGA quando os sensores foram implantados de forma aleatória.

O principal motivo dessa perda de desempenho está relacionada com a técnica determinística usada para definir os vizinhos dos sensores visuais. Essa técnica é mais adequada para ambientes de testes onde os sensores são previamente posicionados de forma igualmente espaçada pela região de monitoramento. Percebe-se que essa abordagem determinística não é a mais adequada para testes com sensores implantados de forma aleatória. Assim, outra abordagem deve ser utilizada na determinação

de vizinhos para esse caso específico a fim de obter resultados mais precisos quando os sensores são implantados de forma aleatória.

Capítulo 5

Conclusão

A melhoria da redundância visual de alvos permite ampliar o escopo de aplicações das redes de sensores visuais sem fio ao permitir a coleta de dados visuais dos alvos monitorados em diferentes perspectivas, aumentando a precisão das informações coletadas. Além disso, oclusões de alvos em determinadas orientações, provocadas por obstáculos diversos, podem ser resolvidas quando a redundância visual é explorada para permitir que mais sensores possam cobrir estes alvos em diferentes perspectivas. A rede de sensores visuais também torna-se mais resistente a falhas uma vez que sensores visuais redundantes podem cobrir um mesmo alvo permitindo que a rede continue operando ainda que alguns sensores falhem. Essas são algumas das importantes vantagens obtidas quando são aplicados algoritmos de maximização da cobertura redundante para as redes de sensores visuais.

Assim, o primeiro algoritmo para maximização da cobertura redundante desenvolvido neste trabalho, nomeado de ECPGA, consiste de uma adaptação do algoritmo CPGA criado por [Costa et al. 2014b]. O CPGA é um algoritmo que já propõe a maximização da redundância visual, dessa forma, a adaptação realizada buscou tão somente otimizá-la para obter resultados melhores tanto de cobertura total dos alvos quanto de aumento da redundância visual. Os testes realizados comprovaram que as adaptações feitas atingiram os objetivos esperados de melhoria de desempenho em relação ao algoritmo original, CPGA.

Contudo, apesar dos bons resultados obtidos com as adaptações desenvolvidas no algoritmo CPGA, tentou-se ainda desenvolver um novo algoritmo usando abordagens diferenciadas em relação as já apresentadas até então na literatura. Assim, foi construído o algoritmo RCMA, que superou as expectativas e conseguiu apresentar resultados superiores em relação aos algoritmos CPGA e ECPGA. Além disso, este algoritmo foi o que apresentou o melhor desempenho em relação aos demais algoritmos de maximização da cobertura redundante desenvolvidos nesse trabalho. O RCMA, apesar de sua abordagem diferenciada, também fez uso das melhorias construídas na adaptação do algoritmo CPGA, mostrando assim a grande importância

da primeira adaptação realizada para se alcançar os expressivos resultados obtidos nesse algoritmo.

Porém, os algoritmos ECPGA e RCMA desenvolvidos são algoritmos centralizados. Nesses algoritmos as decisões são tomadas por um dispositivo central e, em seguida, é repassada a cada sensor da rede a informação sobre a orientação que deve ser assumida. Ou seja, esse tipo de algoritmo deixa toda a responsabilidade a cargo de um dispositivo central, que pode comprometer o funcionamento da rede, quando indisponível por algum motivo. Assim, o último algoritmo desenvolvido foi o algoritmo distribuído nomeado de EDPGA, onde cada nó deve tomar sua própria decisão sobre a melhor orientação a ser adotada. Para isso, cada sensor precisará trocar dados com os demais sensores da rede em busca de informações necessárias para decidir a sua melhor orientação. O desenvolvimento de um algoritmo distribuído com o objetivo de aumentar a redundância visual de alvos em uma rede de sensores visuais, buscando ao mesmo tempo a total cobertura dos alvos, é uma abordagem inovadora e não encontrada na literatura até então. Entretanto, apesar do algoritmo EDPGA ter apresentado resultados superiores aos algoritmos CPGA e ECPGA em relação a cobertura redundante, ele não conseguiu cobrir todos os alvos da aplicação em uma considerável parte dos testes realizados.

Para trabalhos futuros, pretende-se explorar novas abordagens para o desenvolvimento de algoritmos distribuídos no intuito de obter melhores resultados relacionados a cobertura total dos alvos da rede de sensores visuais sem fio. Essa é uma área que merece maior atenção, pois os algoritmos distribuídos tendem a ser mais resistentes a falhas uma vez que não depositam toda a responsabilidade de definir as orientações dos sensores em apenas um nó da rede, como faz os algoritmos centralizados. Pretende-se também, em novos trabalhos, investigar de forma detalhada as vantagens do uso de algoritmos de cobertura com priorização de alvos. Para isso, cenários de testes específicos deverão ser construídos a fim de avaliar o desempenho dos algoritmos desenvolvidos que leva em consideração a prioridade de alvos. Além disso, pretende-se verificar o custo computacional dos algoritmos desenvolvidos para avaliar qual deles apresenta melhor desempenho em termos de consumo de memória e tempo de processamento.

Por fim, percebe-se que a maximização da cobertura redundante nas redes de sensores visuais não apenas traz melhorias relacionadas a disponibilidade da rede e a precisão dos dados, mas permite também que uma série de questões em aberto nessa área de pesquisa sejam solucionados. Esses algoritmos permitem, por exemplo, que mecanismos de segurança à rede de sensores possam ser desenvolvidos para garantir que alvos prioritários em determinada aplicação mantenham-se sempre cobertos por sensores visuais, mesmo sob ataques à segurança. Ou seja, os algoritmos de maximização da cobertura redundantes podem ser usados para fazer com que alvos com oclusões propositais, causadas por pessoas mal intencionadas, possam ser solucionados ao permitir que estes alvos tenham uma cobertura visual redundante. Assim, acredita-se que vários direcionamentos de pesquisa possam surgir conforme mais atenção seja dada a essa área do conhecimento.

Referências Bibliográficas

- [Ai e Abouzeid 2006] Ai, J. e Abouzeid, A. A. (2006). Coverage by directional sensors in randomly deployed wireless sensor networks. *Journal of Combinatorial Optimization*, 11:21–41.
- [Akyildiz et al. 2007] Akyildiz, I. F., Melodia, T., e Chowdhury, K. R. (2007). A survey on wireless multimedia sensor networks. *Computer Networks*, 51:921–960.
- [Akyildiz et al. 2002] Akyildiz, I. F., Su, W., Sankarasubramaniam, Y., e Cayirci, E. (2002). Wireless sensor networks: a survey. *Computer Networks*, 38(4):393–422.
- [Alaei e Barcelo-Ordinas 2009] Alaei, M. e Barcelo-Ordinas, J. M. (2009). A cluster-based scheduling for object detection in wireless multimedia sensor networks. In *Proceedings of the 5th ACM Symposium on QoS and Security for Wireless and Mobile Networks*, Q2SWinet '09, pp. 50–56, New York, NY, USA. ACM.
- [Almalkawi et al. 2010] Almalkawi, I. T., Guerrero Zapata, M., Al-Karaki, J. N., e Morillo-Pozo, J. (2010). Wireless multimedia sensor networks: Current trends and future directions. *Sensors*, 10(7):6662–6717.
- [Baronti et al. 2007] Baronti, P., Pillai, P., Chook, V. W. C., Chessa, S., Gotta, A., e Hu, Y.-F. (2007). Wireless sensor networks: A survey on the state of the art and the 802.15.4 and zigbee standards. *Computer Communications*, 30(7):1655–1695.
- [Boonsongsrikul et al. 2013] Boonsongsrikul, A., Kocijancic, S., e Suppharangsarn, S. (2013). Effective energy consumption on wireless sensor networks: Survey and challenges. In *Information Communication Technology Electronics Microelectronics (MIPRO), 2013 36th International Convention on*, pp. 469–473.
- [Borges et al. 2014] Borges, L., Velez, F., e Lebres, A. (2014). Survey on the characterization and classification of wireless sensor networks applications. *Communications Surveys Tutorials, IEEE*, PP(99):1–1.
- [Chen et al. 2008] Chen, P., Ahammad, P., Boyer, C., Huang, S.-I., Lin, L., Lobaton, E., Meingast, M., Oh, S., Wang, S., Yan, P., Yang, A., Yeo, C., Chang, L.-C., Tygar, J., e Sastry, S. (2008). Citric: A low-bandwidth wireless camera network platform. In *Distributed Smart Cameras, 2008. ICDSC 2008. Second ACM/IEEE International Conference on*, pp. 1–10.

- [Costa 2013] Costa, D. G. (2013). *Otimizações da Transmissão de Imagens em Redes de Sensores Visuais Sem Fio Explorando a Relevância de Monitoramento dos Nós Fontes e Codificação DWT*. PhD thesis, UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE (UFRN), Natal - RN.
- [Costa e Guedes 2010] Costa, D. G. e Guedes, L. A. (2010). The coverage problem in video-based wireless sensor networks: A survey. *Sensors*, 10(9):8215–8247.
- [Costa e Guedes 2011] Costa, D. G. e Guedes, L. A. (2011). A survey on multimedia-based cross-layer optimization in visual sensor networks. *Sensors*, 11(5):5439–5468.
- [Costa e Guedes 2014] Costa, D. G. e Guedes, L. A. (2014). Selecting redundant nodes when addressing availability in wireless visual sensor networks. *IEEE Industrial Informatics*.
- [Costa et al. 2014a] Costa, D. G., Silva, I., Guedes, L. A., Vasques, F., e Portugal, P. (2014a). Availability issues in wireless visual sensor networks. *Sensors*, 14(2):2795–2821.
- [Costa et al. 2014b] Costa, D. G., Silva, I., Guedes, L. A., Vasques, F., e Portugal, P. (2014b). Enhancing redundancy in wireless visual sensor networks for target coverage. *XX Brazilian Symposium on Multimedia and the Web - Webmedia 2014*.
- [Fusco e Gupta 2009] Fusco, G. e Gupta, H. (2009). Selection and orientation of directional sensors for coverage maximization. In *Sensor, Mesh and Ad Hoc Communications and Networks, 2009. SECON '09. 6th Annual IEEE Communications Society Conference on*, pp. 1–9.
- [Garcia-Hernandez et al. 2007] Garcia-Hernandez, C. F., Ibarquengoytia-González, P. H., García-Hernández, J., e Pérez-Díaz, J. A. (2007). Wireless sensor networks and applications: a survey. *International Journal of Computer Science and Network Security*, 17(3):264–273. Survey.
- [Guerrero-Zapata et al. 2010] Guerrero-Zapata, M., Zilan, R., Barceló-Ordinas, J. M., Bicakci, K., e Tavli, B. (2010). The future of security in wireless multimedia sensor networks. *Telecommunication Systems*, 45(1):77–91.
- [Gungor e Hancke 2009] Gungor, V. e Hancke, G. (2009). Industrial wireless sensor networks: Challenges, design principles, and technical approaches. *Industrial Electronics, IEEE Transactions on*, 56(10):4258–4265.
- [Hatanaka et al. 2013] Hatanaka, T., Wasa, Y., e Fujita, M. (2013). Cooperative environmental monitoring for ptz visual sensor networks: A payoff-based learning approach. *Computing Research Repository*, pp. –1–1.
- [Li e Zhang 2013] Li, W. e Zhang, W. (2013). Multiple target localization in wireless visual sensor networks. *Frontiers of Computer Science*, 7(4):496–504.

- [Munishwar et al. 2014] Munishwar, V., Kolar, V., e Abu-Ghazaleh, N. (2014). Coverage in visual sensor networks with pan-tilt-zoom cameras: The maxfov problem. In *INFOCOM, 2014 Proceedings IEEE*, pp. 1492–1500.
- [Munishwar e Abu-Ghazaleh 2013a] Munishwar, V. P. e Abu-Ghazaleh, N. B. (2013a). Coverage algorithms for visual sensor networks. *ACM Transactions on Sensor Networks*, 9(4):45:1–45:36.
- [Munishwar e Abu-Ghazaleh 2013b] Munishwar, V. P. e Abu-Ghazaleh, N. B. (2013b). Coverage algorithms for visual sensor networks. *ACM Transactions on Sensor Networks*, 9(4):45:1–45:36.
- [Rahimi et al. 2005] Rahimi, M., Baer, R., Iroezi, O. I., Garcia, J. C., Warrior, J., Estrin, D., e Srivastava, M. (2005). Cyclops: In situ image sensing and interpretation in wireless sensor networks. In *Proceedings of the 3rd International Conference on Embedded Networked Sensor Systems, SenSys '05*, pp. 192–204, New York, NY, USA. ACM.
- [Tezcan e Wang 2008] Tezcan, N. e Wang, W. (2008). Self-orienting wireless multimedia sensor networks for maximizing multimedia coverage. In *Communications, 2008. ICC '08. IEEE International Conference on*, pp. 2206–2210.
- [Thangaraj e Chitra 2012] Thangaraj, M. e Chitra, P. (2012). Sensors lifetime enhancement techniques in wireless visual sensor networks - a survey. *International Journal of Advanced Research in Computer Engineering and Technology*, 1:6.
- [Yap e Yen 2014] Yap, F. G. H. e Yen, H.-H. (2014). A survey on sensor coverage and visual data capturing/processing/transmission in wireless visual sensor networks. *Sensors*, 14(2):3506–3527.
- [Yen 2013] Yen, H.-H. (2013). Efficient visual sensor coverage algorithm in wireless visual sensor networks. In *Wireless Communications and Mobile Computing Conference (IWCMC), 2013 9th International*, pp. 1516–1521.
- [Yick et al. 2008] Yick, J., Mukherjee, B., e Ghosal, D. (2008). Wireless sensor network survey. *Computer Networks*, 52(12):2292–2330.