

Adaptatividade em Imagens de Satélite para Previsão do Tempo

L. E. C. Dalla Valle and R. L. A. da Rocha

Abstract— Como uma forma alternativa de estudar o clima, com o intuito de melhorar sua previsão usando menos recursos computacionais, um modelo de reconhecimento de padrões em imagens de satélite é abordado. Um gerador de gramática mínima para uma linguagem livre de contexto, cujos símbolos são baseados na própria sequência de imagens é utilizada, e um dispositivo adaptativo é montado sobre esta gramática no intuito de inferir contextos, e, desta forma, prever o comportamento futuro deste sistema.

Keywords— Imagens de Satélite, Adaptatividade.

I. INTRODUÇÃO

DESDE os primeiros indícios de civilização na Antiguidade, as forças da natureza foram motivo de atenção e preocupação. Durante muito tempo, nas sociedades primitivas politeístas, cada evento diferente estava associado a um deus e seus sentimentos em relação à humanidade, afinal, são estas forças que definem períodos de bonança na agricultura, ou a desgraça humana, sendo diretamente associadas com castigos ou premiações. É muito comum os episódios em que Zeus lança raios forjados por seu filho Hefesto, para demonstrar sua ira contra a humanidade. Assim também acontece nas narrativas egípcias, na mitologia nórdica, na crença dos descendentes da raça vermelha e nas tribos africanas.

Como mostra a epígrafe, a bíblia apresenta vários episódios de previsão através de sonhos ou comunicações, como é o caso de Noé, que foi avisado do dilúvio de 40 dias, e José que sonhou com 7 anos de abundância e 7 anos de fome no Egito.

No livro de Jó, há um trecho: *“Com a sua voz tropeja Deus maravilhosamente; faz grandes coisas, que nós não compreendemos. Pois à neve diz: Cai sobre a terra; como também às chuvas e aos aguaceiros: Sede copiosos. Ele sela as mãos de todo homem, para que todos saibam que ele os fez. E as feras entram nos esconderijos e ficam nos seus covis. Da recâmara do sul sai o tufão, e do norte o frio. Ao sopro de Deus forma-se o gelo, e as largas águas são congeladas. Também de umidade carrega as grossas nuvens; as nuvens espalham relâmpagos. Fazem evoluções sob a sua direção, para efetuar tudo quanto lhes ordena sobre a superfície do mundo habitável: seja para disciplina, ou para a sua terra, ou para beneficência, que as faça vir. A isto, Jó, inclina os teus ouvidos; para e considera as obras maravilhosas de Deus. Sabes tu como Deus lhes dá as suas ordens, e faz resplandecer*

o relâmpago da sua nuvem? Compreendes o equilíbrio das nuvens, e as maravilhas daquele que é perfeito nos conhecimentos; tu cujas vestes são quentes, quando há calma sobre a terra por causa do vento sul? Acaso podes, como ele, estender o firmamento, que é sólido como um espelho fundido?...”

O primeiro estudo realizado em meteorologia, foi de Aristóteles, onde, apesar de não contar com instrumentos básicos, como termômetro, barômetro, etc, escreveu o livro Meteorologia, onde descreve seus estudos das nuvens, da chuva, do vento, orvalho, trovão, associa as condições de tempo a estes fenômenos, e foi capaz de realizar algumas previsões precisas sobre as chuvas [6].

Com o passar do tempo, várias descobertas contribuíram para o aprimoramento da técnica. A invenção dos instrumentos, dos

meios de armazenamento e transmissão de dados, a descoberta que a atmosfera poderia ser melhor descrita através de modelos físicos numéricos, a invenção do computador, capaz de realizar inúmeros cálculos em uma pequena fração de tempo, os modelos estatísticos propostos por Lorenz [10], tudo contribuiu para melhorar a qualidade das previsões. Porém nunca mudaram o fato de que existe uma certa imprevisibilidade inerente, e que, como humanidade, continuamos vulneráveis a estas forças.

No Brasil, o Instituto Nacional de Pesquisas Espaciais (INPE), dentre outras atribuições, é o órgão responsável pelos estudos meteorológicos. Em seu Centro de Previsão do Tempo e Estudos Climáticos (CPTEC), os dados das diversas estações meteorológicas espalhadas pelo território nacional são centralizados, e os diversos modelos são processados em supercomputadores para realizar previsões do tempo e clima.

Os modelos mais comuns, processados pelo CPTEC são Global [4,9], Regional (ETA) [2, 12], Ensemble [3, 10] e Oceânico [5, 16]. Todos utilizam simulações numéricas regidas pelas leis físicas e formulações estatística, que representam a realidade aproximada da melhor forma possível dentro do tempo disponível para realizar os cálculos.

O presente trabalho procura investigar, paralelamente, a existência de um novo modelo, não baseado na física, mas na linguagem que a física gera através das formas e movimentações das massas de vapor, no intuito de melhorar o tempo e diminuir a necessidade de processamento, para, com isso, melhorar a qualidade de previsão do tempo.

II. MOTIVAÇÃO

O uso de métodos mais baratos de prever o tempo, e, ao mesmo tempo, que apresentem bons resultados (mesmo quando comparados com aqueles obtidos de modelos

L. E. C. Dalla Valle, Universidade de São Paulo (USP), São Paulo, São Paulo, Brasil, luisdallavalle@usp.br

R. L. A. da Rocha, Universidade de São Paulo (USP), São Paulo, São Paulo, Brasil, rlarocha@usp.br

Agradecemos ao CNPq pelo apoio sem a esta pesquisa.

numéricos que funcionam em super-computadores) é a motivação principal. As questões que impulsionam esta pesquisa são:

- É possível prever o comportamento do clima usando apenas uma sequência de imagens de satélite, com resultados similares aos obtidos com super-computadores?
- É possível inferir alguns dados, como pressão atmosférica, temperatura, umidade, apenas observando estas imagens?



Figura 1: Imagem Original. Fonte: INPE

III. BASES TEÓRICAS

Para a realização deste trabalho, foram necessário estudos em diversas áreas, sendo abordados aqui os principais itens.

A. O Alfabeto

Para tornar possível a utilização do sistema de reescrita e da gramática mínima em uma imagem, um alfabeto bidimensional [Erro! Fonte de referência não encontrada.] baseado em imagem, Σ^{**} foi criado.

A menor unidade de informação de uma imagem é um pixel, no entanto, para uma análise de movimento de padrões, um pixel não é suficiente. A menor unidade mais conveniente é uma matriz 3×3 , pois consegue representar o pixel central e seus vizinhos. Assim, fixando-se um ponto de análise no pixel central, é possível entender o movimento geral daquela região analisando seus vizinhos. Com isso em mente, as imagens criadas para representar os símbolos têm este tamanho, três pixels de largura por três pixels de altura, representados pela matriz:

$$\begin{bmatrix} p_{11} & p_{12} & p_{13} \\ p_{21} & p_{22} & p_{23} \\ p_{31} & p_{32} & p_{33} \end{bmatrix}$$

onde p_{22} é o pixel central de referência.

A imagem que deu origem ao alfabeto, é a imagem resultante do tratamento apresentado na seção **Erro! Fonte de referência não encontrada.**, onde apenas a imagem em preto e branco contendo as bordas das nuvens geradas pelo algoritmo de Canny é utilizada. Consequentemente, o alfabeto é configurado neste mesmo esquema de cores.

Como esta matriz representa uma imagem binária $B : N^2 \rightarrow \{0, 1\}$, e contém ao todo nove pixels, também pode ser representada como uma cadeia de bits da forma $p_{33}p_{32}p_{31}p_{23}p_{22}p_{21}p_{13}p_{12}p_{11}$, sendo possíveis 2^9 combinações diferentes de zeros e uns, totalizando 512 imagens, onde 0 é preto e 1 é branco.

Estas imagens são a representação direta dos símbolos da linguagem criada para esta análise. A Figura **Erro! Fonte de referência não encontrada.** mostra alguns exemplos de símbolos gerados para a linguagem.

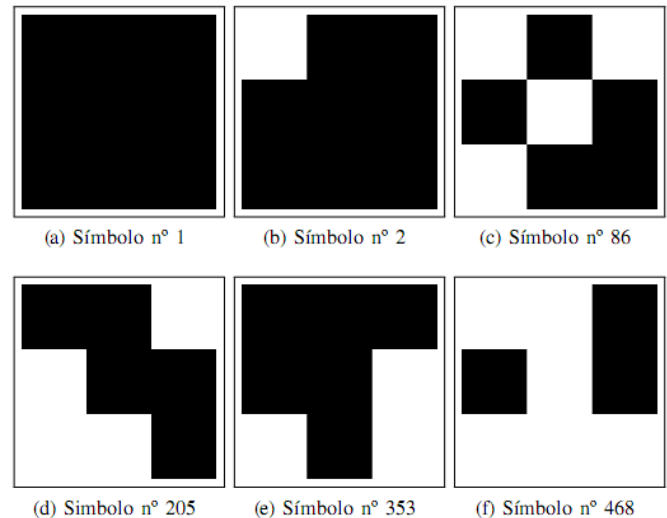


Figura 2: Alguns símbolos criados pela combinação de preto e branco em uma matriz 3×3

B. Autômatos Adaptativos

Autômatos Adaptativos são dispositivos orientados a regras, derivados de Autômatos Finitos, que é um modelo matemático computacional que representa, de forma sintética, um dispositivo mínimo capaz de realizar computação. É constituído de duas partes, uma fita de entrada que contém símbolos, que o autômato está configurado para reconhecer através de uma cabeça de leitura, e o controle finito, uma “caixa preta” que contém todos os estados finais ou não e suas respectivas transições, que verifica o símbolo atual da fita, dada a posição da cabeça de leitura, analisa se o estado atual possui transição para o símbolo lido e transita de estado conforme o caso, repetindo esta operação enquanto houver elementos na fita. Possui três maneiras diferentes de parar. A primeira ocorre quando a cadeia termina e o estado de

transição é final, parando em acerto. A segunda, é quando lê um símbolo cujo estado atual não o reconhece, não existindo transição correspondente, parando em erro. A terceira ocorre quando a fita acaba, e o estado de transição não é final, parando em erro novamente. Segundo **[Erro! Fonte de referência não encontrada.]**, suas utilizações mais importantes são:

1. Software para design e verificação do comportamento de circuitos digitais.
2. O “analisador léxico” de um compilador convencional, que quebra o texto em unidades lógicas, como identificadores, palavras reservadas e pontuação.
3. Software de escaneamento de textos longos, como coleções de páginas web, para encontrar e contabilizar ocorrências de palavras, frases ou outros padrões.
4. Software para verificar sistemas de todos os tipos que possuem número finito de estados, como protocolos de comunicação, ou protocolos para troca segura de informação.

Sua definição matemática segue como apresentado em [13]:

Definição 1: *Um automato finito determinístico é uma quin-tupla $M = (K, \Sigma, \delta, s, F)$, onde:*

- K é um conjunto finito de estados,
- Σ é um alfabeto reconhecido pelo autômato,
- $s \in \Sigma$ é o estado inicial,
- $F \subseteq K$ é o conjunto de estados finais,
- δ é a função de transição, $\delta: K \times \Sigma \rightarrow K$

Existem duas formas de representação de autômatos mais comumente utilizadas, através de grafos (Figura 3) e tabelas de decisão (Tabela 1).

Estado	Entrada	Transição
δ_0	a	δ_1
δ_1	a	δ_1
δ_1	b	δ_0

Tabela 1: Representação em tabela de um autômato finito

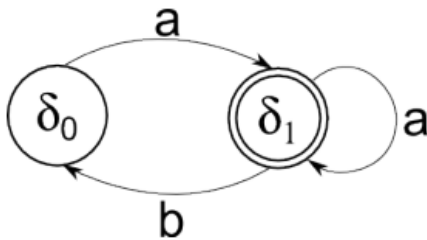


Figura 3: Mesmo autômato da tabela acima, porém representado em um grafo

O autômato finito “determinístico”, apresenta a característica de que cada estado possui no máximo uma transição para um determinado símbolo (não vazio ε) do alfabeto, de forma a não haver ambiguidades em seu comportamento.

Segundo [8], adaptatividade é uma característica apresentada por um dispositivo formal quando este possui a

capacidade de alterar sua própria estrutura interna de forma autônoma, sem interferência externa, nem mesmo de seus usuários, mudando, em consequência, seu próprio comportamento em resposta direta ao estímulo de entrada.

Pode ser aplicada a virtualmente qualquer dispositivo baseado em regras, como tabelas adaptativas e autômatos finitos. Para isto, basta acrescentar à estrutura original do dispositivo, um conjunto de ações adaptativas que serão executadas quando o dispositivo transitar pelos estados. Baseado em [8], um Autômato Adaptativo nada mais é do que a aplicação de um mecanismo adaptativo ao autômato finito convencional definido em III-B

Como visto em [14]:

Definição 2: *Um autômato Adaptativo de Estado Finito (AAF) pode ser definido como uma óctupla $M = (Q, \Sigma, q_0, F, \delta, Q_\infty, \Gamma, \Pi)$, onde:*

$Q \subseteq Q_\infty$ é um conjunto finito de estados

Σ é o alfabeto

q_0 é o estado inicial

$F \subseteq Q$ é o conjunto de estados finais

$\delta \subseteq (Q_\infty \times (\Sigma \cup \{\varepsilon\}) \times Q_\infty)$ é a relação de transição

Q_∞ conjunto teoricamente infinito de estados

$\Gamma \subseteq (\{+, -\} \times (Q_\infty \times (\Sigma \cup \{\varepsilon\}) \times Q_\infty))$ conjunto de ações primitivas de inclusão (+) ou exclusão (−) de transições⁶

$\Pi: (Q_\infty \times (\Sigma \cup \{\varepsilon\}) \times Q_\infty) \rightarrow \Gamma^*$ função que associa a cada transição em δ uma sequência de ações adaptativas primitivas. Pode-se associar uma sequência vazia a uma determinada transição para indicar que é uma transição não adaptativa.

Os autômatos adaptativos são amplamente utilizados em reconhecimento de linguagens, padrões diversos em aprendizado de máquina, etc. Para este trabalho, uma “Camada” adaptativa será utilizada sobre uma gramática mínima gerada, na tentativa de inferir contextos, já que esta, por sua vez, cria uma linguagem livre de contexto.

C. Gramática Mínima

A base de estudo para esta técnica algorítmica foi o trabalho da Ivone Matsuno [11]

Uma gramática é definida em [13] como a quádrupla:

$$G = (V, \Sigma, R, s)$$

$\forall A \in V - \Sigma$ e $u \in V^*$, $A \rightarrow_G u$ quando $(A, u) \in R$

$\forall u, v \in V^*$, onde u, v são sequência de símbolos, e $A \in V - \Sigma | u = xAy, v = xv'v', \text{ e } A \rightarrow_G v'$

Uma gramática é dita livre de contexto, se todas as regras de produção R são da forma $\alpha \rightarrow \beta$, onde $\alpha \in N$ tal que $|\alpha| = 1$ e $\beta = wA$ ou $\beta = w$, e A e $w \in \Sigma^*$

Como definido em [11], uma gramática livre de contexto é dita mínima quando gera uma única sentença. Dada uma sentença x de tamanho n sobre um alfabeto qualquer, a linguagem gerada pela gramática mínima G_x é $L = \{x\}$, e o

⁶ Pelo mesmo motivo apresentado em [14], a operação de consulta “?” é omitida para simplificar a especificação, bem como a ordem em que uma ação adaptativa será executada, se antes ou depois da transição.

problema da gramática mínima é encontrar G_x que gere somente x , de forma que o tamanho de $|G| < n$.

Já uma gramática sensível ao contexto, tem suas regras de produção da forma $\alpha \rightarrow \beta$, onde α é uma palavra de V^*NV^* , β é palavra de V^* e $|\alpha| < |\beta|$, exceto quando $x \rightarrow \varepsilon$ [11]

D. Linguagem Bidimensional

Como forma alternativa de se tratar os símbolos, pode-se utilizar também as técnicas e operações sobre linguagem bidimensional [15]. Que consiste na aplicação das cadeias sobre um alfabeto nas duas dimensões, formando imagens.

Dessa forma, o alfabeto de imagens é tratado como uma linguagem. L é o conjunto de imagens i , tal que, dado o alfabeto $\Sigma = \{0,1\}$, $i \in \Sigma^{**}$ e o tamanho de linhas é igual ao tamanho de colunas $l(p) = c(p) = 3$, permitindo todas as combinações de zeros e uns limitadas nas dimensões 3×3

E. Sistema de Reescrita

O sistema de reescrita aqui utilizado é apenas uma substituição de um símbolo terminal por outro, para realizar a contagem de transições, onde, toda vez que ocorre, o acumulador de transição para este caso é incrementado.

$$S_m \rightarrow S_n \\ Acum_{(m,n)} + +$$

IV. TECNOLOGIAS

Para a implementação desta pesquisa, recorreu-se a algumas tecnologias disponíveis para a implementação algorítmica e interface com o usuário para visualização dos processos.

A. Java

Java é uma Plataforma de Desenvolvimento de Software que inclui a linguagem Java, o compilador, a Máquina Virtual Java, uma extensa biblioteca e um conjunto de ferramentas extras. As grandes vantagens desta plataforma é que é de distribuição e uso livre, e funciona nas principais plataformas computacionais da atualidade, sem exigir reescrita de código ou recompilação para plataformas específicas.

Sua extensa biblioteca oferece classes para criação de interface com o usuário e uma boa quantidade de classes para tratamento e manipulação de imagens utilizadas neste projeto.

B. Python

Durante o desenvolvimento deste projeto, surgiu a necessidade de se alterar a linguagem para Python, por se tratar de uma linguagem que facilita o uso da adaptatividade, por ser dinamicamente modificável em tempo de execução.

Python é uma linguagem funcional, dinâmica, clara, concisa e independente de plataforma, criada em 1990 por Guido van Rossum, no Instituto Nacional de Pesquisa para Matemática e Ciência da Computação da Holanda (CWI).

Pode ser facilmente integrada com outras linguagens e possui uma extensa comunidade de desenvolvedores.

C. QT

QT é uma biblioteca de componentes utilizados em interfaces gráficas de alto nível, multi-plataforma e de grande aceitação pelo mercado. Atualmente é mantida pela Nokia, e é amplamente utilizada em diversos dispositivos móveis e em muitas outras arquiteturas.

Originalmente é desenvolvida em C++, mas possui bibliotecas de ligação (*Bindings*) em Python.

D. Tratamento de Imagens

Como este trabalho envolve exclusivamente a análise de dados a partir de imagens, não poderia deixar de ter um tratamento especial para elas. Um conjunto de algoritmos que aplicam transformações sequenciais, de forma a deixá-la própria para análise.

Estas transformações são comumente chamadas de filtros, pois, como num filtro verdadeiro, retêm substâncias (informações), e deixam passar apenas o que se deseja extrair.

Para este trabalho, uma sequência de três filtros foram aplicados, o mediano, o limiar para conversão em preto e branco, e a detecção de bordas de Canny.

V. O PROCESSO

O sistema criado para realizar esta proposta armazena em memória um conjunto de imagens de satélites copiadas das bases de imagens do INPE. Primeiramente, as linhas dos estados são retiradas, utilizando um filtro mediano, onde as cores de um determinado pixel é calculada usando-se o valor médio dos pixels vizinhos. Como resultado geral, a imagem aparenta-se um pouco mais desfocada, mas não é um problema, já que o processo seguinte, de conversão para preto e branco não é afetado por esta singularidade.

Depois de remover as linhas, a imagem é convertida para preto e branco, utilizando-se um limiar de cores, onde tudo o que é cinza, ou seja, possuem os valores para R, G e B mais elevados e semelhantes, torna-se branco. Todo o resto é convertido para preto. A figura 4 mostra este processo.

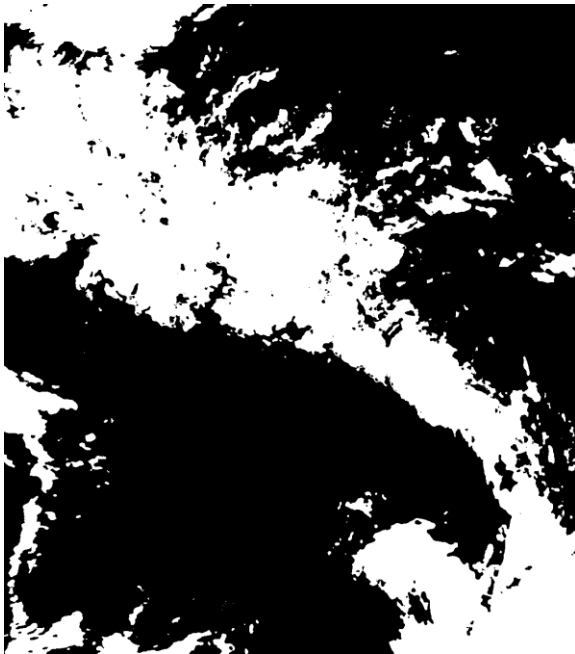


Figura 4: Imagem em P&B

Em seguida, a implementação do algoritmo de Canny [Erro! Fonte de referência não encontrada.] é utilizado, gerando uma borda de um pixel de espessura em todas as nuvens que o processo realizado até o momento detectou, como pode ser visto na Figura Erro! Fonte de referência não encontrada..



Figure 5: Detecção de Bordas usando Canny

Com as imagens preparadas, inicia-se a busca por símbolos e a contagem das transições. Esta busca varre toda a imagem atual e a próxima (caso houver), pixel a pixel, associando

símbolos 3×3, e contando as transições de símbolos de uma imagem para outra.

Os resultados destas contagens são acumulados em uma matriz, como mostra a Tabela **Erro! Fonte de referência não encontrada.** Nesta tabela, as linhas representam os símbolos de origem, e as colunas os símbolos de destino, e as células armazenam a quantidade de transições que ocorreram em uma determinada sequência de imagens.

	Símbolo 0	Símbolo 1	Símbolo 2	...
Símbolo 0	10.498	5.469	4.376	...
Símbolo 1	4.623	3.984	2.445	...
Símbolo 2	4.297	3.243	2.538	...
⋮	⋮	⋮	⋮	⋮

Tabela 2: Representação da Matriz de Acumulação de Transições, onde a linha representa o símbolo de origem e a coluna o símbolo de destino

Para representar de forma mais abrangente as transições de todos os símbolos, tomou-se emprestado o *micro-array* da genética, mudando sua denominação para *macro-array*, que, para este caso, segue a mesma formação da Tabela **Erro! Fonte de referência não encontrada.**, como mostram as Figuras **Erro! Fonte de referência não encontrada.** e **Erro! Fonte de referência não encontrada.**, onde as cores dos pixels⁷ são definidas pela quantidade de transições (n), através de uma simples conversão de base, da forma:

$$COR(Simb_i, Simb_j) = \begin{cases} R = mod(n, 256) \\ G = mod(div(n, 256), 256) \\ B = mod(div(n, 256^2), 256) \end{cases}$$

Obviamente que com essa representação, não podem haver mais que 256^3 transições, o que é um número muito além do necessário para o estado atual da pesquisa. Este processo permite duas formas distintas de análise. Uma se baseia no acúmulo de imagens da sequência, como é o caso da Figura **Erro! Fonte de referência não encontrada.**, que acumulou as transições das primeiras 60 imagens analisadas, apresentando cores mais intensas. Outra forma é realizar a análise duas a duas, onde um conjunto de macro-arrays é gerado no modelo da Figura **Erro! Fonte de referência não encontrada.**, acumulando o resultado da análise da primeira imagem com a segunda, em seguida da segunda com a terceira, e assim sucessivamente, permitindo a criação de uma animação de curta duração destes arrays.

Ambas são úteis, pois permitem analisar as transições que ocorreram em períodos determinados, como, por exemplo, o verão deste ano comparando ao verão do ano passado,

⁷ Por motivos de impressão, as imagens do macro-array tiveram seu contraste e saturação alterados, de forma a tornar mais simples a interpretação visual dos dados no texto impresso

períodos mais curtos, como a manhã de ontem com a manhã de hoje, ou mesmo fenômenos conhecidos, como comparar o *El-niño* do ano passado e o *El-niño* deste ano.

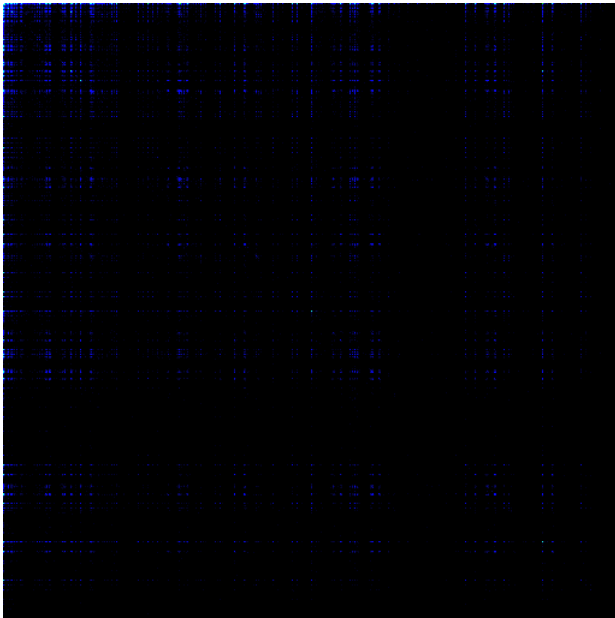


Figure 6: Match para duas imagens

Com estas transições armazenadas, um gerador de gramática mínima entra em ação, verificando onde ocorreram as transições, e em seguida, agrupa as repetições de forma a reduzir (compactar) os dados obtidos. Este processo baseia-se no algoritmo LZ77 [**Erro! Fonte de referência não encontrada.**] para realizar a tarefa.

A gramática gerada pelo processo descrito acima é, por definição, livre de contexto. Porém, é perceptível que a linguagem que rege a movimentação aqui analisada é notavelmente sensível ao contexto, e talvez, em alguns casos até irrestrita.

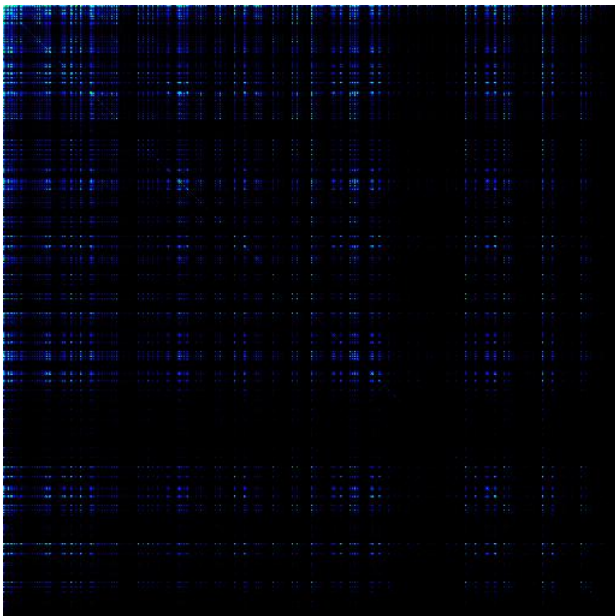


Figure 7: Result of Match para sessenta imagens

Para inferir contextos, um dispositivo adaptativo é montado sobre a gramática gerada, de forma a gerar um conjunto de regras bem definidas, às quais o sistema pode se basear para realizar a previsão dos eventos futuros.

VI. ANÁLISES

É possível notar nas Figuras **Erro! Fonte de referência não encontrada.** e **Erro! Fonte de referência não encontrada.**, que as primeiras linhas e colunas possuem grande quantidade de transições, por apresentarem cores mais intensas. Isto se deve ao fato de que a maior parte das imagens analisadas é composta de preto, sendo correspondente ao símbolo nº1, e tanto a transição deste símbolo para ele mesmo, como dele para os outros e de outros para ele, são as que ocorrem com maior frequência, o que implica em sua possível remoção em estudos futuros, pois apenas os símbolos de significância maior em correspondência às bordas detectadas serão mantidos.

É facilmente notada também uma componente diagonal, do pixel (0,0) até o (512,512), o que indica ausência de transição. Este fato se deve ao movimento lento de algumas bordas analisadas, da correspondência ocasional do mesmo símbolo por puro acaso, ou por movimentos circulatorios padronizados da atmosfera. Esta componente diagonal tenderá a ficar menos evidente com a melhor calibração do modelo.

É importante salientar que há uma dissipação de ocorrências dos primeiros símbolos para os últimos. Aparentemente pelo fato de que as bordas se correspondem melhor com os primeiros símbolos, que possuem mais pixels pretos do que brancos, formando padrões lineares, do que os últimos, que formam massas densas de branco.

Ao final, um grafo é montado contendo os símbolos em seus vértices e os pesos das derivações em suas arestas, onde o tamanho da aresta é inversamente proporcional ao peso que ela contém, para aproximar os vértices com maior peso de transições e afastar os mais leves, no intuito de adquirir uma resposta visual do processo realizado.

VII. CONCLUSÃO

No atual estado da pesquisa, o processo está modelado. O sistema de *match*, para detectar os símbolos nas imagens de satélite está desenvolvido, e contabilizando as transições, exibindo os resultados nos *macro-arrays*.

Entre as tarefas pendentes, estão a migração para Python, criação do gerador de gramática mínima, utilizando o algoritmo LZ77, definir como as imagens resultado servirão de base para a gramática, se é que essa forma de acúmulo será útil para a geração da linguagem livre de contexto, e aplicar a camada adaptativa para inferir contextos.

Na próxima fase do trabalho, partindo de uma imagem de satélite, será utilizada a gramática encontrada e os contextos inferidos para montar as próximas imagens e calcular o quanto elas refletem os acontecimentos futuros.

REFERÊNCIAS

- [1] C. de Previsão do Tempo e Estudos Climáticos (CPTEC-INPE), “Cptec website.” www.cptec.inpe.br, Junho 2010. Acesso em 8 de junho de 2010.
- [2] E. N. Lorenz, “A study of the predictability of a 28-variable atmospheric model,” *Tellus*, vol. 17, p. 321–333, 1965. doi: 10.1111/j.2153-3490.1965.tb01424.x.
- [3] CPTEC, “Modelo global.” http://previsaonumerica.cptec.inpe.br/mod_glb.shtml, Setembro 2010.
- [4] J. L. Kinter, D. DeWitt, P. A. Dirmeyer, M. J. Fennessy, B. P. Kirtman, L. Marx, S. Edwin K, J. Shukla, and D. M. Strauss, “The climate atmosphere-biosphere general circulation model. volume1: Formulation,” tech. rep., Center for Ocean-Land-Atmosphere Studies., Calverton, USA., 1997. Report n.o 51.
- [5] CPTEC, “Eta model.” <http://etamodel.cptec.inpe.br/>, Setembro 2010.
- [6] F. Mesinger, “Eta model at ncep: Challenges overcome and lessons learned. lecture notes,” in *Workshop on “Design and Use of Regional Weather Prediction Models”*, (Miramare, Trieste, Italy), p. 42 pp, The Abdus Salam International Centre for Theoretical Physics, 11-19 April 2005.
- [7] CPTEC, “Modelo ensemble.” http://previsaonumerica.cptec.inpe.br/mod_ens.shtml, Setembro 2010.
- [8] CPTEC, “Modelo wwatch.” http://previsaonumerica.cptec.inpe.br/mod_wwatch.shtml, Setembro 2010.
- [9] H. L. Tolman, “The numerical model wavewatch: a third generation model for the hindcasting of wind waves on tides in shelf seas,” *Communications on Hydraulic and Geotechnical Engineering, Delft Univ. of Techn.*, ISSN 0169-6548, vol. 89-2, p. 72 pp., 1989.
- [10] G. Rozenberg and A. Saloma, eds., *Handbook of Formal Languages*, vol. 3. Springer, 1997.
- [11] J. E. Hopcroft, R. Motwani, and J. D. Ullman, *Finite Automata*. Pearson Education and Addison-Wesley, 2001.
- [12] C. H. Papadimitriou and H. R. Lewis, *Elements of the Theory of Computation*. Prentice-Hall, 1998.
- [13] J. José Neto, “Adaptive rule-driven devices - general formulation and case study,” *Lecture Notes in Computer Science*, vol. 2494/2002, pp. 466–470, 2002.
- [14] H. Pistori and J. J. Neto, “Adaptree - proposta de um algoritmo para indução de Árvores de decisão baseado em técnicas adaptativas,” *Anais Conferência Latino Americana de Informática*, vol. Novembro/2002, pp. 1–10, 2002.
- [15] I. P. Matsuno, “Um estudo dos processos de inferência de gramáticas regulares e livres de contexto baseado em modelos adaptativos,” Master’s thesis, Universidade de São Paulo, 2006.
- [16] J. Canny, “A computational approach to edge detection,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 8, pp. 679 – 698, 1986.